



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**VYHLEDÁVÁNÍ DUPLICITNÍCH FOTOGRAFIÍ**

SEARCH FOR DUPLICITIES OF PHOTOS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ZDENĚK SKLENÁŘ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Prof. Dr. Ing. PAVEL ZEMČÍK**

**BRNO 2018**

## Zadání bakalářské práce

Student: **Sklenář Zdeněk**  
Program: Informační technologie  
Název: **Vyhledávání duplicitních fotografií**  
**Search for Duplicities of Photos**

Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte literaturu na téma informací obsažených v digitálních fotografiích a existující software pro vyhledávání duplicit se zaměřením na hledání podle doplňkových údajů fotografií.
2. Navrhněte postup vyhledávání duplicit ve fotografiích podle doplňkových informací ve fotografiích a navrhněte i uživatelské rozhraní pro volbu vhodné fotografie z duplicitních, zvažte i jednoduché metriky kvality podle obsahu fotografií.
3. Zvolte implementační prostředí a způsob efektivní implementace navrženého postupu vyhledávání i uživatelského rozhraní.
4. Implementujte vyhledávání i uživatelské rozhraní a demonstруйте funkčnost na vhodném příkladu uživatelského scénáře.
5. Zhodnoťte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a popis jednotlivých etap řešení. Odevzdává se v elektronické podobě a ve dvou výtiscích, přičemž oba musí obsahovat podepsané prohlášení o autorství. Jeden výtisk musí být svázan nerozebíratelným způsobem.

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**  
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.  
Datum zadání: 11. července 2018  
Datum odevzdání: 31. července 2018

## Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem, implementací a otestováním aplikace, která slouží k vyhledání duplikací ve fotografiích podle Exif metadat. Aplikace dále umožňuje zobrazení náhledu fotografie, včetně Exif metadat fotografie. Seskupení duplikací, následný výběr nejlepší fotografie pro ponechání podle parametru nastaveného uživatelem, manuální úpravu této volby a vymazání ostatních. Dále je možné filtrování fotografií a export vybraných fotografií do ZIP archivu.

## Abstract

This bachelor thesis is about the analysis, design, implementation and testing of an application which is used to find duplicates in photographs according to its Exif metadata. The application also enables preview of photos, including Exif metadata. Additionally it is possible to filter photos, group duplicities with the original photo, and select the best photos to keep it in accordance with a user-defined parameter, then manually adjust this option, and deleting others. It is also a possible to export selected photos to a ZIP archive.

## Klíčová slova

Digitální fotografie, Metadata, Exif, IPTC, Duplicita, Python3, pyQT5, cx\_Freeze, export, Kvalita fotografie, ZIP

## Keywords

Digital photo, Metadata, Exif, IPTC, Duplicate, Python3, pyQT5, cx\_Freeze, export, Photo quality, ZIP

## Citace

SKLENÁŘ, Zdeněk. *Vyhledávání duplicitních fotografií*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

# Vyhledávání duplicitních fotografií

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Zdeněk Sklenář  
31.07.2018

## Poděkování

Rád bych poděkoval hlavně svému vedoucímu práce panu Prof. Dr. Ing. Pavlovi Zemčíkovi za pevné nervy a jeho věcné připomínky k práci. Také své rodině a přátelům, kteří dokázali vidět naději v momentech kdy jsem to já nedokázal, bez jejich podpory by tato práce nikdy nevznikla. V neposlední řadě také všem, kteří mě naučili to, co znám a nezlomili nade mnou hůl, popřípadě ochotně pořídili novou.

Zvláštní poděkování patří odbornému konzultantovi panu Ing. Martinovi Čachovi. Děkuji za jeho úhel pohledu na tuto práci a jeho čas, který na konzultace vymezil.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Digitální fotografie</b>	<b>2</b>
2.1	Kvalita digitálních fotografií . . . . .	3
2.2	Formáty digitálních fotografií . . . . .	4
2.3	Archivace digitálních fotografií . . . . .	5
2.4	Metadata . . . . .	6
2.5	Duplicita ve fotografiích . . . . .	9
<b>3</b>	<b>Existující řešení</b>	<b>11</b>
3.1	Anti-Twin . . . . .	11
3.2	ExifTool . . . . .	12
3.3	Duplicate Photos Fixer Pro . . . . .	13
3.4	Visual Similarity Duplicate Images Finder . . . . .	14
3.5	VisiPics . . . . .	15
<b>4</b>	<b>Návrh</b>	<b>16</b>
4.1	Analýza požadavků . . . . .	16
4.2	Scénáře použití aplikace . . . . .	18
4.3	Návrh realizace . . . . .	18
4.4	Uživatelské rozhraní . . . . .	20
<b>5</b>	<b>Implementace</b>	<b>22</b>
5.1	Grafické uživatelské rozhraní . . . . .	23
5.2	Vyhledání duplicit . . . . .	24
5.3	Vyhodnocení kvality fotografie . . . . .	27
5.4	Realizace vláken . . . . .	28
<b>6</b>	<b>Vyhodnocení aplikace</b>	<b>29</b>
6.1	Automatické testování . . . . .	30
6.2	Porovnání s konkurencí . . . . .	32
6.3	Možná rozšíření aplikace . . . . .	32
6.4	Shrnutí výsledků . . . . .	32
<b>7</b>	<b>Závěr</b>	<b>33</b>
	<b>Literatura</b>	<b>34</b>
<b>A</b>	<b>Obsah přiloženého média</b>	<b>37</b>

# Seznam obrázků

2.1	Detail fotografie [19]	2
2.2	Schéma ZIP archivu, který obsahuje dva soubory [25]	6
2.3	Ukázka Exif metadat fotografie	7
2.4	Detail fotografie s IPTC metadaty [6]	8
2.5	Ukázka duplicitních fotografií včetně jejich metadat	10
3.1	Ukázka vyhledání duplicit v aplikaci AntiTwin [24]	11
3.2	Ukázka výpisu metadat fotografie pomocí ExifTool	12
3.3	Ukázka zobrazení duplicit v aplikaci Duplicate Photos Fixer Pro [5]	13
3.4	Ukázka zobrazení duplicit v aplikaci Duplicate Photos Finder [3]	14
3.5	Ukázka aplikace Visipics [11]	15
4.1	První čtyři nejvíce oblíbené jazyky, podle PYPL žebříčku [9]	17
4.2	Obecný vývojový diagram aplikace	19
4.3	Náčrt základního okna aplikace	20
4.4	Náčrt okna duplicit aplikace	20
4.5	Náčrt okna s informacemi o fotografii	21
4.6	Náčrt okna s nastavením aplikace	21
5.1	Schéma modulů aplikace	22
5.2	Ukázka zobrazení duplicitních skupin	23
5.3	Obecný vývojový diagram vyhledávání duplicit v načtených fotografiích	24
5.4	Ukázka manuálního porovnání duplicit	27
5.5	Náhled vláken aplikace	28
6.1	Graf výsledků 2.běhu automatických testů	31

# Kapitola 1

## Úvod

V dnešní digitální době je snadné během krátkého času vytvořit velké množství fotografií. Všechny tyto fotografie se ale musí někam ukládat a často tak ve velkém počtu zaplňují naše úložiště. Proto vznikla snaha o jejich třídění a určení duplicitních fotografií, které je zbytečné uchovávat. Jejich vymazáním tak vznikne místo pro další, důležitější fotografie. Až tedy budete příště kupovat větší úložné médium pro svoji dlouholetou galerii, zamyslete se nejdříve nad tím, zda opravdu potřebujete všechny její položky.

Cílem této práce bylo nastudovat teorii k problematice vyhledání duplicity ve fotografiích podle Exif metadat, a poté vytvořit aplikaci pro jejich detekci a vymazání. Jako volitelné cíle byly možnosti zobrazení metadat a jejich úprava, jednotlivá i hromadná. Také určení kvality fotografie tak, aby měl uživatel snazší výběr fotografie, kterou si ponechá.

Aplikace by měla pomoci běžnému uživateli protřídit jeho dlouhodobé úložiště fotografií ve kterém se pravděpodobně nachází duplicity. Je zde například více verzí fotografie, protože byla postupem času několikrát upravována. Popřípadě můžou být tyto verze i naprosto totožné, ať už z důvodů zálohy nebo výběru do jiné složky. Aplikace tyto stejné a upravené verze fotografie seskupí a nabídne uživateli náhled na jednotlivé skupiny duplicit. Určení duplicity tedy není nijak závislé na obrazových datech fotografie.

Tuto práci jsem si vybral, protože už jsem se osobně několikrát potýkal s touto problematikou u své vlastní galerie a zajímaly mne tedy způsoby, jak ji efektivně pročistit.

Tato bakalářská práce se skládá ze shrnutí současného stavu popsaného v kapitolách 2, 3. Specifikování požadavků, návrhu grafického uživatelského rozhraní v kapitole 4. Popis implementace aplikace v kapitole 5. Následuje otestování vlastností této implementace na reálných datech a jejich vyhodnocení v kapitole 6. Tato kapitola obsahuje také porovnání s konkurencí.

## Kapitola 2

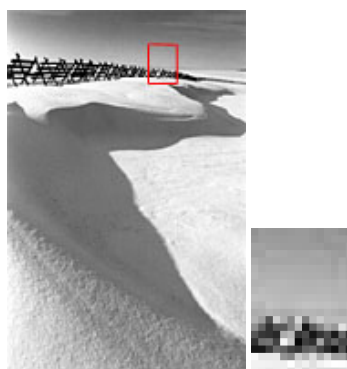
# Digitální fotografie

V této kapitole je čtenář obeznámen se základní historií a principy digitální fotografie, které jsou podstatné pro pochopení této bakalářské práce. Více informací se může čtenář dozvědět zde [22].

Klasická fotografie provází lidstvo již od předminulého století, prožila velký vývoj a stále se ještě vyvíjí. I když nehrozí, že by digitální fotografie zcela nahradila klasický film, trend posledních let ukazuje, že si stále více lidí pořizuje digitální fotoaparáty [36]. Rozdíly mezi digitální a klasickou fotografií jsou odpovědí na otázku vzniku a expanze do mnoha oblastí.

Hlavními důvody rozšíření digitálních fotografií je rychlost pořízení fotografie a také snadná přenositelnost. Další výhodou je možnost zkontrolování výsledné fotografie ihned po pořízení a není potřeba, tak jak je to u klasické fotografie, fotografie zdlouhavě vyvolávat [36]. Nespornou výhodou je i nízká cena pořízení jedné fotografie. I když pořizovací cena digitálního fotoaparátu je vyšší, tak lze na něj v podstatě zadarmo pořizovat nespočet fotografií. Z toho vyplývá i kapacita přístrojů. Při pořizování klasických fotografií je potřeba využití filmu s kapacitou přibližně 36 snímků. Počet digitálních fotografií je omezen pouze kapacitou paměťového média.

Zjednodušeně lze říci, že digitální fotografii je možné si představit jako mozaiku, skládající se z jednotlivých čtverečků tedy pixelů. Pixel je nejmenší element, který tvoří digitální obraz a nemá pevně danou velikost. Síť vytvářející pixely se nazývá rastr. Velikost pixelu v rastru výrazně ovlivňuje kvalitu fotografie, určuje totiž i jejich počet v rastru [22]. Čím je síť hustší, tím ostřejší obraz lze získat. Rastr digitálního obrazu se objeví v okamžiku, kdy obraz dostatečně zvětšíme.



Obrázek 2.1: Detail fotografie [19]



## 2.1 Kvalita digitálních fotografií

Kvalita fotografie je přímo závislá na snímacím zařízení. Kolik detailů je schopno zachytit, kolik světla a barev, jaký má šum, jakou má stabilizaci, jak rychle je schopný fotografii zachytit a s jakou mírou komprimace do příslušného formátu zapsat [26]. Všechny tyto parametry se výrazně podepisují na kvalitě výsledného snímku. Konkrétní případy jsou uvedeny dále.

### 2.1.1 Fotoaparát

Každý digitální fotoaparát je složen z několika elektronických součástí, jejichž účelem je převod analogové informace přenášené fotony na informaci digitální a poté případná komprese obrazových dat na vhodný formát. Hlavní částí procesu je získání obrazu, který má fotoaparát zachytit. Fotony dopadající na obrazový snímač fotoaparátu nesou informaci, která je převáděna na analogový signál, který je následně pomocí analogově číslicového převodníku převáděna na signál digitální [17]. Následně prochází snímek dalšími zpracovávacími procesy, jako například korekce barev, nebo vyvážení kontrastu [22]. Nakonec je zkomprimován do požadovaného formátu a uložen na zapisovací médium fotoaparátu.

### 2.1.2 Ostrost

Ostrost ve fotografii je hranový kontrast obrazu. Lidskému oku se zdají kontrastnější obrázky ostřejší, závisí tudíž na velikosti derivace kontrastu vzhledem k prostoru. Při fotografování se fotoaparát zaostřuje pohybem čočky (nebo více čoček) v objektivu. Správné zaostření zajistí, že na snímku bude ostré to, co si fotograf ze scény vybere [22]. Nastavením zaostřovací vzdálenosti fotograf určí tzv. rovinu zaostření. Objekty v této rovině jsou zobrazeny ostře, objekty před a za ní jsou ostré méně. Míru této neostrosti neboli hloubku ostrosti ovlivňuje nastavení clony na fotoaparátu.

### 2.1.3 Barevnost

Neboli barevná teplota fotografie. Závisí na intenzitě barev, při větší intenzitě barev dosáhneme teplejších barev. Naopak při menší intenzitě barev bude mít fotografie studenější barvy. Při pořizování fotografie se dá tato vlastnost fotografie ovlivnit vyvážením bílé barvy ve fotoaparátu, popřípadě použitím barevných filtrů, které lze přidělat k přední části objektivu. Záleží tedy vždy na aktuálním osvětlení, při kterém je fotografie pořizována a na kalibraci fotoaparátu [22]. Na základě určení bílé barvy je pak vyvážena intenzita ostatních barev. Černobílá fotografie má nulovou barevnost.

### 2.1.4 Šum

Digitální fotoaparát je obvod složený z elektrických součástí, každá tato součástka zanechá do zachycené fotografie určitou míru šumu, kterému říkáme náhodný, tedy svoji chybovost. Kromě náhodného šumu lze sledovat také šum temný nebo zesilovací [17]. Temný šum nejčastěji vzniká při dlouhém expozičním čase<sup>1</sup>, ale jeho příčinou může být také rostoucí teplota nebo stárí fotoaparátu. Zesilovací šum je způsoben zvýšením hodnoty citlivosti, tzv. ISO<sup>2</sup>, za účelem zesvětlení fotografie při použití stejného expozičního času i clony.

<sup>1</sup>Expoziční čas - čas, po který na snímek nebo na čip proudí světlo při pořízení fotografie

<sup>2</sup>ISO - hodnota citlivosti při pořízení snímku

Existuje ještě spousta dalších vlastností fotografie, které ovlivňují její kvalitu. Stejně tak se nedá s jistotou říci, že například fotografie s teplejšími barvami bude vždy kvalitnější, nicméně se dá stanovit vzorec, který bude platit pro většinu případů [10].

## 2.2 Formáty digitálních fotografií

Jak již bylo zmíněno dříve, tak jedním z nejdůležitějších parametrů pro uchování digitální fotografie je formát souboru, do kterého je uložena. Formát může výrazně ovlivnit například kvalitu a velikost fotografie [15]. Existuje mnoho formátů fotografií, které se liší například v úrovni komprese nebo velikosti. Nejběžnější formáty pro uložení digitálních fotografií jsou popsány dále.

### JPEG

(Joint Photographic Experts Group) [32]

Jedná se o nejrozšířenější formát digitálních fotografií a obrázků vůbec. Používá standardní metodu ztrátové komprese používané pro ukládání počítačových obrázků ve foto-realistické kvalitě. Tento formát se hodí na fotografie, u kterých se díky ztrátové kompresi dá ušetřit hodně datového objemu a lehká ztráta kvality při běžném pozorování příliš nevadí [15].

Formát byl navržen speciálně pro fotografie, takže většinou nenabízí uspokojivou kvalitu, pokud je obsahem něco jiného než obraz. Pokud je ale potřeba kvalitní fotografie, je tento formát vhodný spíše pro konečný export upravené fotografie. Díky své malé velikosti je často používán na webu [32].

### GIF

(Graphics Interchange Format)

GIF je grafický formát určený pro rastrovou grafiku. Používá bezeztrátovou kompresi LZW (bezeztrátový algoritmus), na rozdíl například od formátu JPEG, který používá ztrátovou kompresi. GIF je tedy vhodný pro reprezentaci vektorových obrázků (nápis, plánky, loga), protože zakreslené čáry nejsou rozmazány kompresí. Tento formát umožňuje také jednoduché animace [31], na rozdíl od formátu PNG.

GIF má jedno velké omezení — maximální počet současně použitých barev barevné palety je 256 (8 bitů), v případě animace pak umožňuje využít odlišné palety 256 barev pro každý snímek. Toto omezení nemá formát PNG, který se hodí ke stejným účelům jako GIF a nabízí pro většinu obrazů výrazně lepší kompresi.

### PNG

(Portable Network Graphics)

Jedná se o grafický formát určený pro bezeztrátovou kompresi rastrové grafiky. Byl vytvořen jako zdokonalení a náhrada formátu GIF, který byl patentově chráněný, dnes jsou patenty prošlé. PNG nabízí podporu 24 bitové barevné hloubky, nemá tedy jako GIF omezení na maximální počet 256 barev současně. PNG tedy do jisté míry nahrazuje GIF, nabízí více barev a lepší kompresi. Navíc obsahuje osmibitovou průhlednost (tzv. alfa kanál). To znamená, že obrázek může být v různých částech různě průhledný (tzv. RGBA barevný model). Nevýhodou PNG oproti GIF je, že nepodporuje animace [33]. Je vhodný typicky pro webovou grafiku. Nevýhoda je vysoká datová velikost oproti JPG při ukládání fotografií.

## RAW

(z anglického slova raw, což znamená v překladu surový, nezpracovaný)

Jedná se o soubor obsahující minimálně zpracovaná data ze snímače digitálního fotoaparátu. RAW není přímo souborový formát, ale spíše třída (či klasifikace) souborových formátů, protože každý výrobce implementuje jiný formát RAW souborů. Každý z RAW formátů má svoji příponu v názvu souborů, podle které je lze identifikovat. RAW soubory jsou digitální obdobou negativů, tzn. RAW soubor není přímo použitelný jako obrázek, ale obsahuje všechny potřebné informace k jeho vytvoření [34].

RAW formát je preferován především profesionálními fotografy a nadšenci, protože umožňuje větší možnosti bezztrátových úprav než formáty JPEG. Mezi tyto úpravy patří především korekce expozice, vyvážení bílé, úprava tonality apod. Ovšem ani formáty RAW nemusejí být bezztrátové. Některé formáty jsou již z chipu uloženy v redukované bitové hloubce [34].

## TIFF

(Tagged Image File Format)

Formát TIFF tvoří neoficiální standard pro ukládání snímků určených pro tisk. TIFF je komplikovanější formát oproti ostatním formátům pro ukládání rastrové grafiky. Tento formát vytvořila společnost Aldus v roce 1986. TIFF umožňuje jako jeden z mála grafických formátů vícestránkové soubory, a proto se často používá například pro ukládání přijatých faxů přijatých pomocí počítače [35].

TIFF je tzv. kontejnerový formát neboli formát, který dokáže nést různá obrazová data komprimovaná různými kodeky. Kontejner je potom jen standardní přenašeč opatřený všemi důležitými visačkami, které vnitřní formát správně identifikují a popisují. Díky tomu je TIFF velmi flexibilní formát, který dokáže kombinovat různé obrázky uvnitř jednoho souboru, dokáže používat různé barevné hloubky, různé způsoby komprese atp. [23].

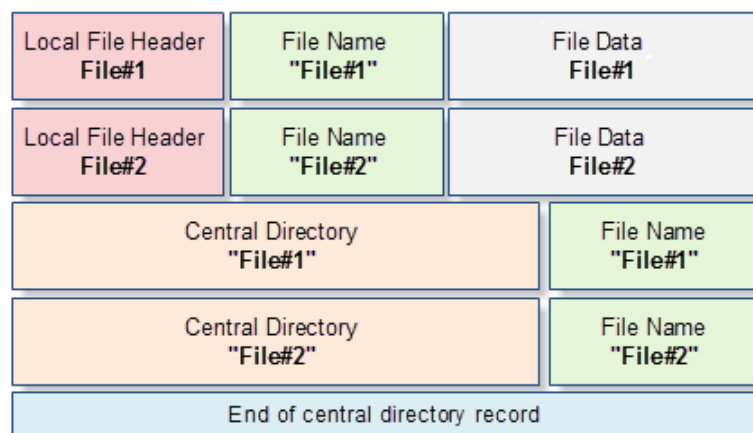
Na rozdíl od formátu JPEG nabízí možnost bezztrátové komprese, vyšší barevné hloubky a průhlednosti. Dnes již sláva tohoto formátu poněkud opadla, protože byl na straně fotoaparátů nahrazen formátem RAW [23].

## 2.3 Archivace digitálních fotografií

Umožňuje vybrané digitální fotografie uschovat do jednoho souboru, za účelem zmenšení velikosti výsledného souboru než samostatných fotografií díky bezztrátové kompresi. Může sloužit jako zálohování určité části velkých galerií, kterou už nechceme přesouvat jako celek. Metadata fotografií jsou při jejich archivaci zachována, jedná se tedy o bezpečný způsob uložení zpracovaných fotografií i pro jejich pozdější opětovné použití.

### 2.3.1 ZIP

Jedná se o jeden z prvních formátů souborových archivů, který podporoval kompresi dat. Specifikace formátu .ZIP byla publikována pod veřejnou doménou. Tímto krokem pomohl formátu stát se celosvětovým otevřeným standardem. Archiv by měl obsahovat alespoň jeden nebo více souborů, také může obsahovat složky [2]. Tento formát byl zvolen hlavně pro svoji jednoduchost a volitelnost komprese. Také má vestavěnou podporu ve většině operačních systémů.



Obrázek 2.2: Schéma ZIP archivu, který obsahuje dva soubory [25]

Obsah archivu lze pro lepší orientaci rozdělit na část obsahující definice a data uložených souborů a na část reprezentující organizaci adresářů a souborů [29].

- První část obsahuje záznamy, jež se opakují pro každý uložený soubor. Jeden takovýto záznam musí obsahovat alespoň lokální hlavičku souboru (Local File Header) a data souboru (File Data), většinou obsahuje také název souboru (File Name). Pokud je nastaven třetí bit položky General purpose bit flag v lokální hlavičce souboru, musíme ještě počítat s tím, že za data byla přidána sekce Data description o velikosti 12 bajtů.
- Druhá část se skládá ze struktury hlaviček centrálního adresáře (Central Directory Header). Počet těchto hlaviček odpovídá počtu adresářů a souborů obsažených v tomto archivu. Hlavička centrálního adresáře začíná signaturou [0x50, 0x4b, 0x01, 0x02], podle které je v souboru identifikovatelný začátek hlavičky. Za ní následují metadata souboru, např.: datum a čas poslední úpravy obsaženého souboru, velikost před a po kompresi atd. Tato struktura hlaviček je ukončena položkou Záznam konce centrálního adresáře (End of central directory record). Ten začíná signaturou [0x50, 0x4b, 0x05, 0x06] a obsahuje informace o tom na kterém disku je soubor uložen, na kterém disku začíná struktura centrálního adresáře a další.

Existuje ještě spousta dalších formátů archivace [29], nicméně pro účely této bakalářské práce stačí informace o formátu ZIP.

## 2.4 Metadata

Metadata jsou skupina textových informací o datech. Metadata ve fotografiích umožňují zachovat informace, které jsou přenášeny s obrazovými daty.

Způsob uložení metadat si definuje každý formát zvlášť. Metadata zapisuje fotoaparát při pořízení snímku. Zapiše například datum, GPS polohu, svůj typ a značku výrobce. V dnešní době sociálních sítí tak vzrůstala obava z informací, které by se daly z fotografií vyčíst. Proto sociální sítě promazávají například GPS údaje z metadat fotografie [18]. Postupem času vzniklo několik formátů popsaných dále.

### 2.4.1 EXIF

(Exchangeable image file format)

Je specifikace pro formát metadat, vkládaných do souborů digitálními fotoaparáty. Informace se vkládají do existujících souborových formátů, jako je JPEG, TIFF revize 6.0, PNG a RIFF WAVE. Struktura EXIF dat je převzatá ze souborového formátu TIFF. Datové standardy typu TIFF, Exif, TIFF/EP a DCF si jsou velice podobné. Z grafických formátů nepodporuje GIF. Také nepodporuje video formáty. Informace jsou ukládány interně v určené části souboru. Exif navrhla japonská průmyslové asociace JEITA, verze 2.1 vznikla 21. června 1998, verze 2.2 v dubnu 2002 a verze 2.31 v dubnu 2010 [30].

Metadata v Exifu mohou mimo jiné obsahovat: značku a model fotoaparátu, datum a čas pořízení snímku, nastavení fotoaparátu (citlivost, clona, expoziční čas, ohnisková vzdálenost, informace o použití blesku atd.), náhled snímku, informace o místu pořízení (GPS souřadnice), komentáře a informace o autorovi [18]. Největší výhodou tohoto formátu je jeho rozsáhlé používání u výrobců fotoaparátů. Téměř každá fotografie dnes má svoje Exif metadata. Na základě, kterých se o ní lze dozvědět další informace.

```
{
  'Make': 'LENOVO',
  'Model': 'Lenovo S1',
  'Orientation': '1',
  'XResolution': '(72, 1)',
  'YResolution': '(72, 1)',
  'GPSInfo': '16: 'M', 17: (0, 1)',
  'ResolutionUnit': '2',
  'Software': 'Lenovo Camera',
  'DateTime': '2016:06:24 17:06:24',
  'ColorSpace': '1',
  'ExifImageWidth': '2560',
  'LightSource': '255',
  'FocalLength': '(350, 100)',
  'ExifImageHeight': '1920',
  'Flash': '0',
  'ExifInteroperabilityOffset': '812',
  'ISOSpeedRatings': '91',
  'DigitalZoomRatio': '(100, 100)'
}
```

Obrázek 2.3: Ukázka Exif metadat fotografie

Z uvedeného příkladu Exif metadat fotografie můžeme například vyčíst:

- výrobce snímacího zařízení (Lenovo)
- přesný typ/model snímacího zařízení (Lenovo S1)
- přesné datum a čas pořízení fotografie (24.06.2016 17:06:24)
- šířku a výšku fotografie (2560x1920)
- název software/programu který fotografii vytvořil (Lenovo Camera)

Na základě těchto informací můžeme třeba najít upravené fotografie, sestavit časový diagram fotografií, sestavit trasu pohybu fotografa apod.

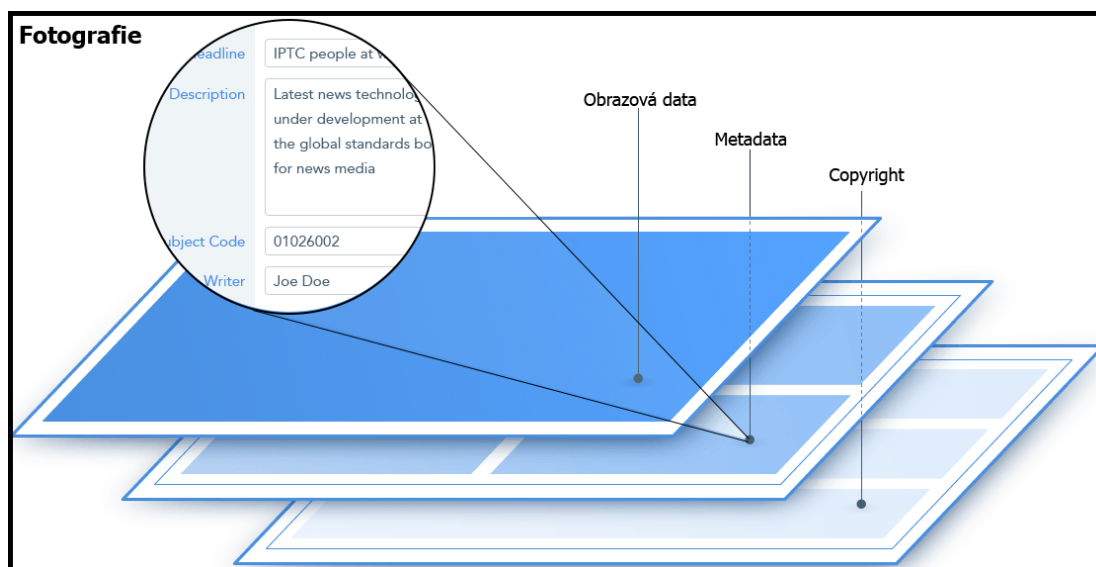
## 2.4.2 IPTC

(International Press Telecommunications Council)

Jsou textová data, která se, podobně jako metadata Exif, využívají pro vkládání informací přímo do grafických souborů jpeg a tiff a jsou rozdělena do dvou částí [6]. První je pojmenována Iptc Core shema, druhá pak Iptc Extension shema, která rozšiřuje část první.

Na rozdíl od metadat Exif, které se týkaly především nastavení, výrobce a modelu fotoaparátu, kterým jsou také vkládány, se metadata Iptc týkají především autora fotografie a samotného obsahu fotografie a bývají přidávána dodatečně [18]. Umožňují uložit adresu, kontakt, autorská práva, titulky, klíčová slova, která jsou velmi využívána při vyhledávání, kategorie, do které snímek spadá a mnoho dalšího.

Podporuje všechny druhy médií například text, fotografie, video i zvuk. Informace jsou uloženy interně v určené části souboru [6]. Význam tohoto formátu je v současnosti na vzestupu.



Obrázek 2.4: Detail fotografie s IPTC metadaty [6]

## 2.4.3 XMP

(Adobe Extensible Metadata Platform)

Vychází ze značovacího jazyka XML. Metadata jsou ukládána do samostatného souboru, který je přidružen k multimediálnímu souboru. Open-source standard byl vytvořen v roce 2001 společností Adobe Systems Inc. [1].

Formát XMP definuje vlastnosti, které jsou užitečné pro získání informací o vzniku a editaci fotografie. Základními daty, které je možné uchovávat v XMP jsou: popis, autor, předmět fotografie, vydavatel, datum, typ, formát apod. Díky rozšiřitelnosti XMP si pak další data může uživatel přidávat sám. Výhodou, oproti předchozím typům metadat, je větší podpora typů souborů, která není omezena na typy jpeg a tiff [18].

## 2.5 Duplicita ve fotografiích

Duplicita fotografií může být určena několika způsoby. Hodně ale záleží na tom, co vlastně považujeme za duplicitu. Pokud totiž budeme prohledávat dlouholetou galerii uživatele, určitě nalezneme duplicitu úplnou, tedy tu stejnou fotografii/soubor jenom v jiném umístění. Upravenou verzi fotografie. Několik hodně podobných fotografií z krátkého časového úseku, kdy se uživatel třeba snažil vyfotit svoji rodinu bez toho, aby někdo mrknul. Všechny tyto fotografie lze považovat za duplicitní, neboť uživatel potřebuje pouze 1 či 2 z těchto fotografií z 10, které má ve své galerii.

### 2.5.1 Určení duplicity podle metadat

Tato metoda porovnává fotografie na základě textových údajů, které vytvořilo snímací zařízení při jejich vzniku. Lze porovnávat záznam po záznamu anebo spočítat hash pro všechny záznamy u obou fotografií a tyto hash pak porovnat [27]. V obou případech bude ale porovnání velice rychlé. Pokud budeme porovnávat záznam po záznamu lze třeba udělat časový interval, kdy jsou fotografie duplicitní. Uživatel totiž nemůže být schopný udělat dvě fotografie, které se datem pořízení liší o jednu vteřinu a nejsou duplicitní. Každý uživatel bude mít ale tento maximální čas duplicity trochu jiný, proto je potřeba aby zde byl nastavitelný interval. Stejně tak platí, že tento způsob porovnání nalezne i jakkoliv upravené fotografie a seskupí je s originálem, dokud uživatel nebo aplikace nesmaže či neupraví klíčová metadata. Přesnost této metody závisí na jejím nastavení.

### Částečná duplicita metadat

V tomto případě není duplicita určena všemi záznamy. Pro vyhledání duplicit je použit složený klíč. Složený klíč je složen ze záznamů, podle kterého lze identifikovat duplicitu. Pokud tedy dvě fotografie mají stejné datum pořízení a stejné snímací zařízení které je pořídilo lze je označit jako duplicitní.

### 2.5.2 Určení duplicity podle bitové shody

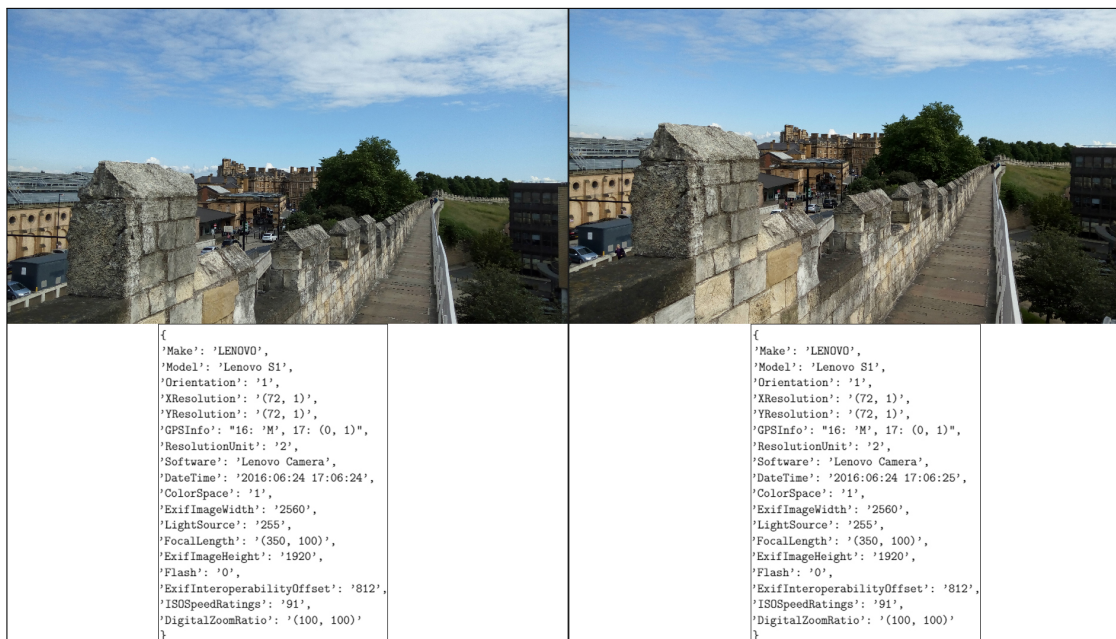
(byte-to-byte)

Tato metoda najde pouze úplně stejné fotografie. Stačí pouze jakkoliv malá změna a duplicita není nalezena. Jedná se o hodně přesnou metodu pro nalezení úplných duplicit. Princip spočívá v porovnání souborů. Bez ohledu na obsah je binárně porovnána hodnota každého bytu. Velmi jednoduchá implementace a střední časová náročnost. Může být vylepšeno procentuální shodou a tresholdem.

### 2.5.3 Určení duplicity podle obrazových dat

Tato metoda má tu výhodu, že určuje duplicitu podle stejných dat jako uživatel a je tedy velice přesná. Nenajde více upravené fotografie. Je k dispozici několik možných variant [13]. Obtížná implementace, časová náročnost závisí na použité metodě, je ale určitě větší než u porovnání metadat. Může být vylepšeno procentuální shodou a tresholdem.





Obrázek 2.5: Ukázka duplicitních fotografií včetně jejich metadat

Fotografie jsou z hlediska obrazových dat téměř totožné. Podle metadat můžeme zjistit, že tyto fotografie byly pořízeny pouze jednu sekundu po sobě, také jsou vyfoceny na stejném místě, se stejnou orientací snímacího zařízení atd. Je tedy velice pravděpodobné, že jsou duplicitní. Tuto duplicitu lze s jistotou určit podle obrazu i metadat.

V této kapitole byly shrnuty základní vlastnosti digitálních fotografií, základní metriky pro určení jejich kvality, možnost jejich archivace a způsoby určení duplicity. V následující kapitole je využito těchto znalostí k průzkumu existujících řešení.



## Kapitola 3

# Existující řešení

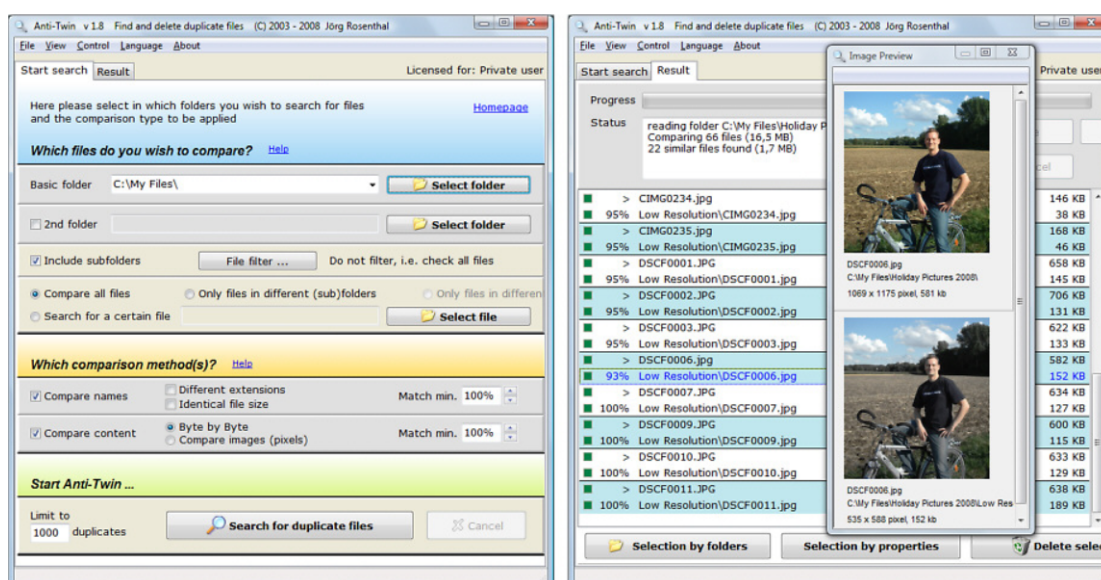
V rámci shrnutí aktuálního stavu, byly vyhledány aplikace, které se zabývají problematikou vyhledání duplicit ve fotografiích. Bylo zpracováno několik článků [4], [16] a vhodní zástupci jsou uvedeni dále.

### 3.1 Anti-Twin

Přímo podporuje pouze platformu Windows. Nepřímo přes nástroj Wine také Linux a Mac. Aplikace je k dostání pod volnou licenci pro soukromé použití.

Pro určení duplicity používá porovnání typu byte-to-byte nebo pixel-to-pixel. Také lze porovnávat podle obsahu a názvu souboru. Při porovnání lze také vzít v úvahu název a velikost souborů. Nalezené duplicity lze vymazat do koše nebo úplně. Lze zadat pouze dvě zdrojové složky, pro jedno vyhledávání. Umí prohledávat i podsložky [24].

Lze použít nejen na vyhledání duplicit ve fotografiích. Ale třeba i na hudební soubory MP3. Všechny údaje uvedené v této práci platí pro verzi 1.8d.



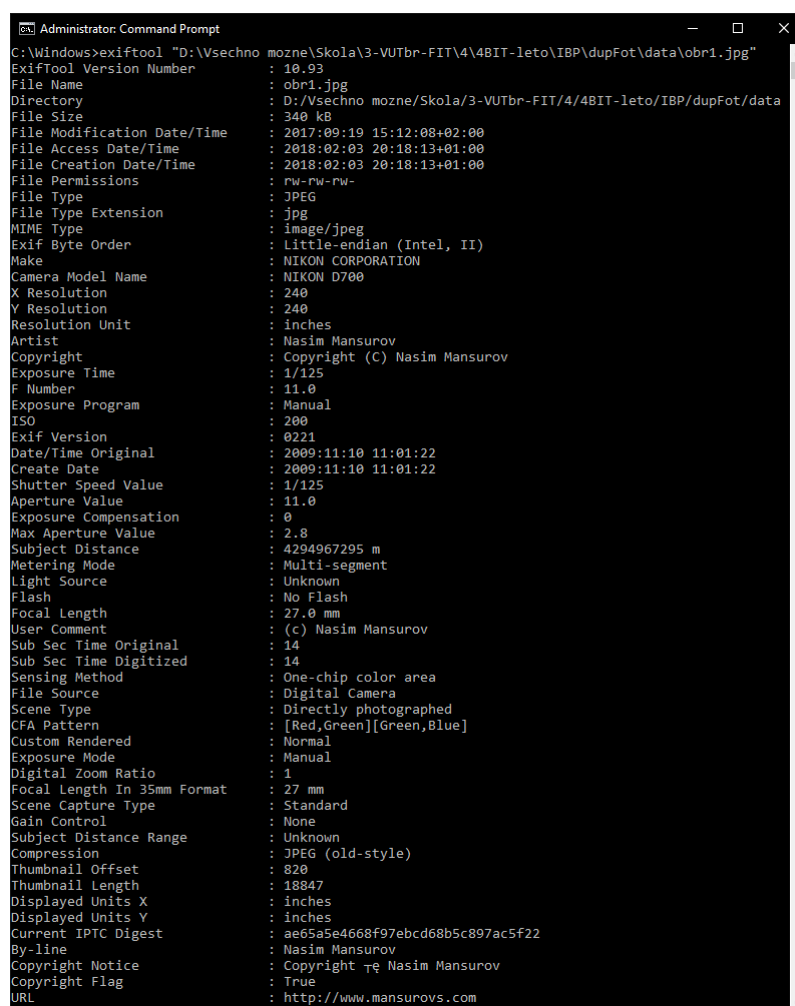
Obrázek 3.1: Ukázka vyhledání duplicit v aplikaci AntiTwin [24]

## 3.2 ExifTool

Podporuje Windows, Linux a Mac platformy. Pouze konzolová aplikace bez GUI. Lze také použít jako knihovnu pro další projekty. Rozsáhlá internetová dokumentace [12]. Možnost číst a upravovat Exif, IPTC, XMP metadata. Hromadně nebo pro skupinu fotografií. Má více jazykových verzí. Podporuje téměř všechny multimediální formáty. Dlouhodobě testováno na fotografiích z více jak tisíc typů zařízení. Nelze bez dalších rozšíření použít pro vyhledání duplicitních Exif metadat fotografií.

Na vytvoření GUI už vznikl samostatný projekt pyExifToolGui [14], který ale podle slov autora není ani zdaleka dokončen.

Aplikace i projekt na grafickou nástavbu je poskytován zdarma. Nástavba má dostupné zdrojové kódy. Zdrojové kódy pro aplikaci nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 10.94.



```
Administrator: Command Prompt
C:\Windows>exiftool "D:\Vsechno mozne\Skola\3-VUTbr-FIT\4\4BIT-let0\IBP\dupFot\data\obr1.jpg"
ExifTool Version Number      : 10.93
File Name                    : obr1.jpg
Directory                   : D:\Vsechno mozne\Skola\3-VUTbr-FIT\4\4BIT-let0\IBP\dupFot\data
File Size                    : 340 KB
File Modification Date/Time  : 2017:09:19 15:12:08+02:00
File Access Date/Time       : 2018:02:03 20:18:13+01:00
File Creation Date/Time     : 2018:02:03 20:18:13+01:00
File Permissions             : rw-rw-rw-
File Type                    : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
Exif Byte Order              : Little-endian (Intel, II)
Make                        : NIKON CORPORATION
Camera Model Name           : NIKON D700
X Resolution                 : 240
Y Resolution                 : 240
Resolution Unit              : inches
Artist                      : Nasim Mansurov
Copyright                   : Copyright (C) Nasim Mansurov
Exposure Time                : 1/125
F Number                     : 11.0
Exposure Program             : Manual
ISO                          : 200
Exif Version                 : 0221
Date/Time Original          : 2009:11:10 11:01:22
Create Date                  : 2009:11:10 11:01:22
Shutter Speed Value          : 1/125
Aperture Value               : 11.0
Exposure Compensation        : 0
Max Aperture Value           : 2.8
Subject Distance             : 4204967295 m
Metering Mode                : Multi-segment
Light Source                 : Unknown
Flash                        : No Flash
Focal Length                 : 27.0 mm
User Comment                 : (c) Nasim Mansurov
Sub Sec Time Original        : 14
Sub Sec Time Digitized       : 14
Sensing Method               : One-chip color area
File Source                  : Digital Camera
Scene Type                   : Directly photographed
CFA Pattern                  : [Red,Green][Green,Blue]
Custom Rendered              : Normal
Exposure Mode                : Manual
Digital Zoom Ratio           : 1
Focal Length In 35mm Format  : 27 mm
Scene Capture Type           : Standard
Gain Control                 : None
Subject Distance Range       : Unknown
Compression                  : JPEG (old-style)
Thumbnail Offset             : 820
Thumbnail Length             : 18847
Displayed Units X             : inches
Displayed Units Y            : inches
Current IPTC Digest          : ae65a5e4668f97ebcd68b5c897ac5f22
By-line                      : Nasim Mansurov
Copyright Notice              : Copyright © Nasim Mansurov
Copyright Flag                : True
URL                           : http://www.mansurovs.com
```

Obrázek 3.2: Ukázka výpisu metadat fotografie pomocí ExifTool

### 3.3 Duplicate Photos Fixer Pro

Podporuje Windows, iOS, Mac a Android platformy. Aplikace má moderní, tmavé grafické rozhraní. Obsahuje překlad do 14 jazyků, čeština bohužel chybí.

Pro určení duplicity používá volitelný interval shody GPS a datumu pořízení fotografie [5]. Také volitelnou míru shody obrazových dat. Seskupí fotografie do skupin duplicit a nabídne jejich souhrnný náhled. S automatickým výběrem nejvhodnější pro ponechání. Je možná manuální úprava volby. Smaže označené fotografie.

Plná verze není poskytována zdarma. Lze aplikaci vyzkoušet v Trial verzi, která ale nedovolí vymazání vybraných fotografií. Zdrojové kódy nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 1.1.1086.5815.

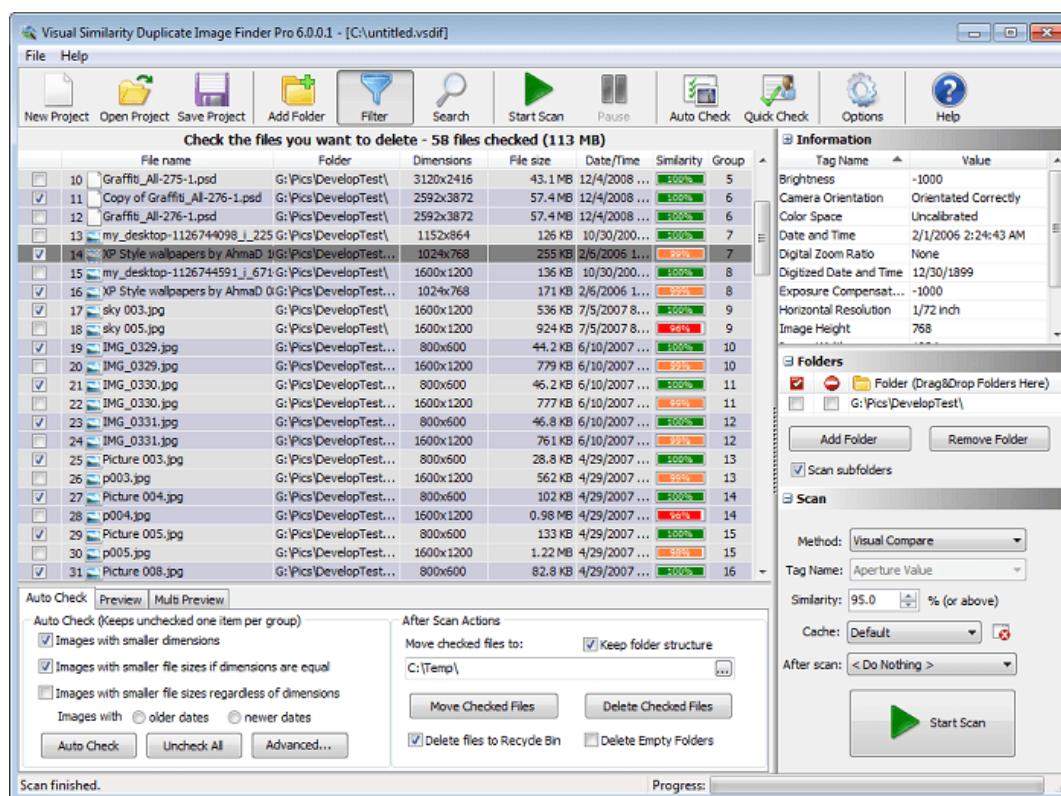


Obrázek 3.3: Ukázka zobrazení duplicit v aplikaci Duplicate Photos Fixer Pro [5]

### 3.4 Visual Similarity Duplicate Images Finder

Podporuje pouze platformu Windows. Umožňuje nastavit procentuální shodu duplicity. Jejich seskupení a automatický výběr nejvhodnější fotografie pro ponechání. Mezi metodami pro určení duplicity si lze vybrat mezi: Obraz, Exif datum (lze nastavit interval), volitelný Exif záznam, velikost souboru, hash [3]. Lze nastavit procentuální shodu. Umí zobrazit Exif metadata u fotografie. Označené fotografie lze smazat do koše, smazat úplně, kopírovat, přesunout. Projekt lze uložit a načíst později.

Plná verze aplikace není poskytována zdarma. Lze stáhnout pouze velice omezenou Demo verzi. Všechny údaje uvedené v této práci platí pro verzi Demo 6.7.0.1.

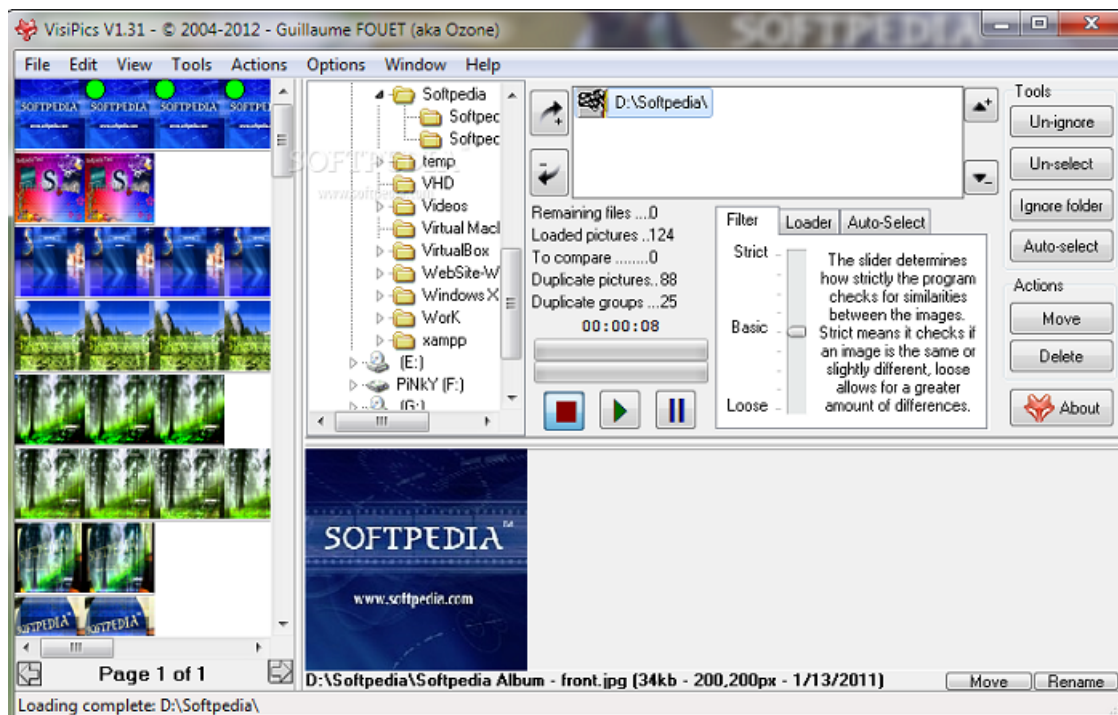


Obrázek 3.4: Ukázka zobrazení duplicit v aplikaci Duplicate Photos Finder [3]

### 3.5 VisiPics

Podporuje pouze Windows platformu. Funkce automatického výběru fotografie pro ponechání ze skupiny duplicit. Hromadné zobrazení skupin duplicit. Duplicity určuje podle obrazových dat. Duplicity mohou mít rozdílné rozlišení, jiný formát, minimální barevné efekty [11]. Aplikace má tři módy pro vyhledání duplicity. Pro nastavení tolerance velikosti odlišností fotografií.

Aplikace je poskytována v plné verzi zdarma. Zdrojové kódy nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 1.31.



Obrázek 3.5: Ukázka aplikace Visipics [11]

Získané zkušenosti z nastudování a zkoušení existujících řešení byly dále využity v návrhu aplikace. Tento výběr bude také složit pro srovnání mé aplikace s nejbližší konkurencí.

Název aplikace	převzaté funkce
Anti-Twin	nastavitelné zahrnutí podsložek, možnost vymazání duplicit do koše
ExifTool	zobrazení metadat, hromadná editace metadat
VS DIF Demo	skupiny duplicit, automatický výběr nejlepší fotografie pro ponechání
DP Fixer Pro	časový interval pro vyhledání duplicit, přesunutí výběru fotografií
VisiPics	více módů vyhledání duplicity
Společné vlastnosti	platforma Windows, podpora JPEG formátu

Tabulka 3.1: Převzaté funkce od existujících řešení

# Kapitola 4

## Návrh

V této kapitole je uveden postup návrhu aplikace. Na začátku je nutné zvolit technologie pro její realizaci a její funkce na základě požadavků od cílové skupiny. Jako souhrn požadavků od cílové skupiny posloužilo zadání práce a jeho doplnění, upřesnění na konzultacích s vedoucím práce. Zcela záměrně je zde použita velká úroveň abstrakce, která je objasněna v implementaci.

### 4.1 Analýza požadavků

Ze souhrnu požadavků a zpracování existujících řešení vyplynuly následující funkce:

- Aplikace najde a setřídí duplicity do skupin. V těchto skupinách provede aplikace automatický výběr a poté umožní uživateli vybrat, které si chce ponechat. Více typů vyhledání duplicity.
- Zobrazení metadat fotografie.
- Zobrazení ignorovaných fotografií bez metadat a umožnit jejich smazání.
- Omezený výběr a následný bezztrátový export fotografií na základě zadanych filtrů.
- Duplicita je určena podle Exif metadat.
- Volitelné zahrnutí podsložek při procházení zdrojů od uživatele u každého zdroje.
- Možnost vymazání duplicit do koše nebo úplně.
- Při určení duplicity lze nastavit velikost časové mezery kde duplicita končí.

Rozhodl jsem se aplikaci realizovat hlavně pro platformu Windows, a to pro verzi 7 a 10. Jde o nejrozšířenější platformu mezi uživateli, aplikace tak bude mít co nejvíce potenciálních uživatelů. Sekundární je pak podpora operačních systémů typu Linux a Mac. Hlavní je podpora JPEG formátu, protože většina digitálních fotoaparátů vytváří fotografie právě v tomto formátu.



### 4.1.1 Výběr implementačního prostředí

Jednou z prvních a nejdůležitějších věcí je volba jazyka ve kterém bude celá aplikace napsána. Zvolený jazyk musí obsahovat všechny funkce, aby v něm vytvořená aplikace byla schopna splňovat všechny definované požadavky.

## Python

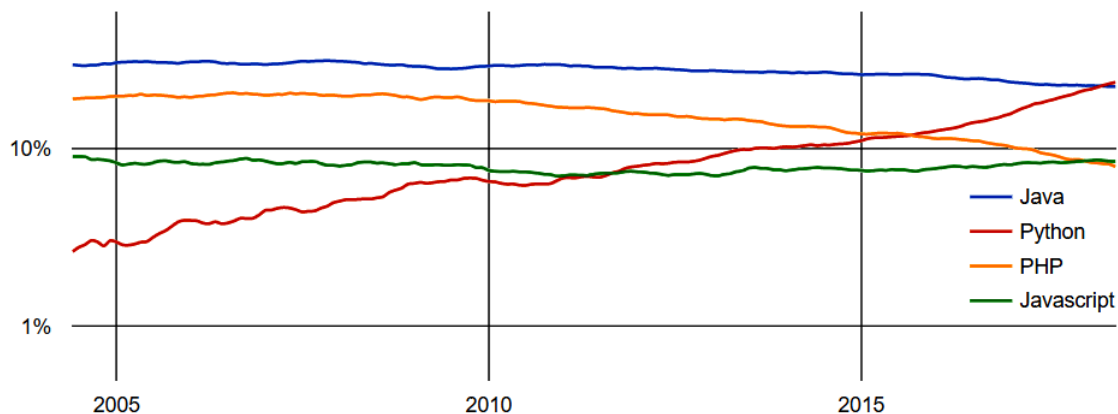
Existuje spousta článků zabývajících se tím, zda Python patří mezi skriptovací jazyky. Určitě má z této oblasti mnoho vlastností, ale dnes je samotnými tvůrci označován jako interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou [7]. Je tedy vhodný pro skriptování, vývoj webových aplikací, vývoj GUI aplikací, vědecké a numerické výpočty, softwarový vývoj nebo systémovou administraci.

Nepatří mezi silně typované programovací jazyky. Je vyvíjen jako open source. Používá systém výjimek, které lze za použití bloku try-except zachytit a ošetřit. Umožňuje vícenásobnou dědičnost.

Klade důraz na produktivitu programátora. Má jednoduché konstrukce a dobrou čitelnost programu. Pro oddělení bloků programu jsou zde použity hlavně tzv. bílé znaky (mezera, tabulátor, konec řádku), hloubka zanoření odpovídá počtu odsazení bloku.

Má velkou zásobu snadno použitelných knihoven včetně jejich dokumentace. Pro správné pochopení jazyka je také dobré znát několik zásad od jeho tvůrců. [21] Je třeba zkompilovat program pro každý typ operačního systému zvlášť. Verze 2.x není kompatibilní s verzí 3.x.

Popularita jazyka je dlouhodobě na vzestupu. Podle internetových žebříčků, například poněkud nevyrovnaný TIOBE žebříček [8] nebo níže ukázaný PYPL žebříček [9]. Na zmíněném odkazu se i ve FAQ lze dozvědět porovnání technik obou žebříčků.



Obrázek 4.1: První čtyři nejvíce oblíbené jazyky, podle PYPL žebříčku [9]

Nároky na jazyk byly dány hlavně zadáním, bylo tedy nutné vytvořit GUI pro interakci s uživatelem tzv. frontend, který bude využívat funkce z backendu. Nejlépe těmto požadavkům vyhovoval právě tento jazyk. Proto byl pro implementaci zvolen jazyk Python, a to ve verzi 3.6.

## 4.2 Scénáře použití aplikace

Aplikace půjde použít několika způsoby. Ne všichni uživatelé totiž potřebují aplikaci ke stejnému použití, proto je potřeba aplikaci navrhnout tak, aby ji mohli používat různí uživatelé různými způsoby. V rámci návrhu aplikace, byly tedy uvažovány tyto případy použití:

- První počítá s možností opatrného a podrobného přístupu. Uživatel zadá všechny zdrojové složky sám tzn. bez procházení podsložek. Nastaví možnost automatického výběru a mazání do koše, aby v případě chyby nepřišel o fotografii úplně. Rozhodně si chce také zobrazit podrobné údaje o každé fotografii. Při upravování výběru fotografií pro ponechání. Také mít možnost detailního porovnání fotografií mezi kterými se rozhoduje. Chce vidět upozornění, že fotografie budou smazány. Na konci si překontroluje jejich počet.
- Druhý je zaměřen spíše na rychlý přístup k pročištění galerie. Uživatel chce zadat pouze jednu složku a zahrnout všechny její podsložky automaticky. Popřípadě přidat další složku rychleji než tu první. Do nastavení se nedívat opakovaně, pouze při prvním spuštění aplikace. Zobrazit a smazat ignorované fotografie. Neupravit automatický výběr a okamžitě smazat fotografie nevybrané pro ponechání. Popřípadě postupně promazávat jednotlivé skupiny. Potvrdit a prohlédnout si jejich počet.
- Třetí scénář je zaměřen na chybně nastavené datum snímacího zařízení při pořízení fotografie. Uživatel zadá zdrojové složky. Poté zvolí hromadnou editaci metadat. Ze zobrazených fotografií zvolí u kterých chce datum posunout a o jak velký časový úsek. Potvrdí provedení změn.
- Poslední scénář je zaměřen na zálohu fotografií. Uživatel zadá zdrojové složky. Zaškrtně možnost podsložek. Nastaví preferenci pro ponechání a zobrazení i neduplicitních fotografií. Zadá filtry pro vyhledání fotografií. Poté proběhne automatický výběr nejvhodnějších fotografií pro ponechání podle nastaveného parametru. Uživatel zvolí export, zadá název a umístění archivu, pokud se mu nevyhovují výchozí hodnoty. Výsledný archiv obsahuje filtrované fotografie bez duplicit v původní adresářové struktuře.

Všechny tyto scénáře lze využít při testování aplikace uživateli, také slouží jako seznam případů užití.

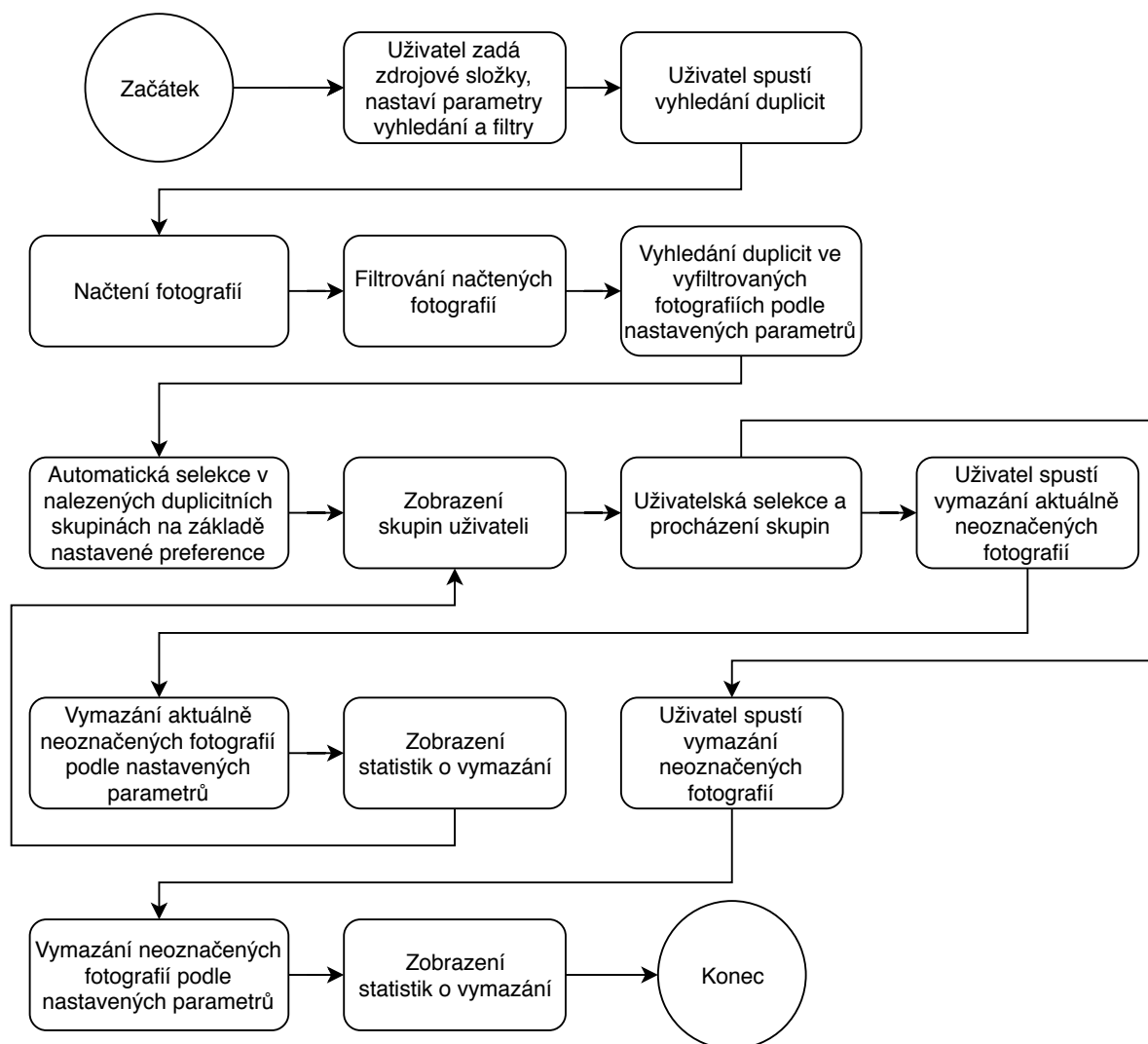
## 4.3 Návrh realizace

Vyplývá z kapitoly 4.1, aplikace má za úkol načíst fotografie ze zadaných zdrojů a vyhledat v nich duplicity podle metadat. Na základě vstupního nastavení od uživatele má být možné změnit některé klíčové parametry vyhledávání, protože každá galerie bude o trochu jiná a uživatel by tak měl dostat možnost nastavit aplikaci na míru té svojí, pro dosažení co nejlepších výsledků. Aplikace nesmí přestat reagovat, naopak musí uživateli dát zpětnou vazbu v jakékoliv fázi zpracovávání. Může být spuštěno pouze jedno prohledávání v jeden okamžik. Je tedy nutné zablokovat příslušná tlačítka uživateli. Důležitá je také stabilita a správnost určení duplicity. To zajistí více vláknová implementace aplikace.



### 4.3.1 Jednotlivé části aplikace

Jednotlivé části běhu aplikace a jejich pořadí lze vidět níže.



Obrázek 4.2: Obecný vývojový diagram aplikace

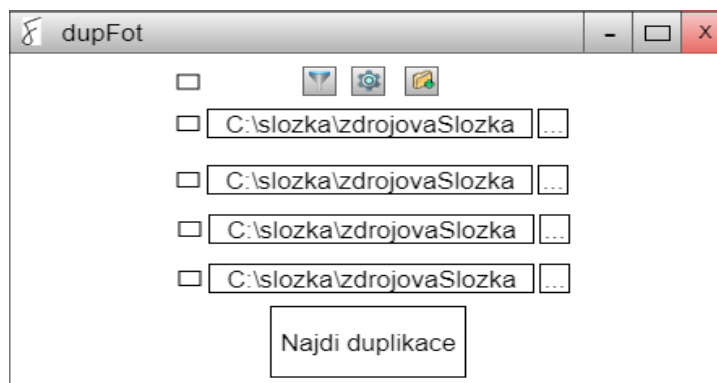
Každá tato část tvoří samostatný celek, akci, která bude v průběhu aplikace vykonána. Jednotlivé části mezi sebou budou propojeny funkcí, která zajistí jejich postupné provedení a předávání jejich výstupů.

## 4.4 Uživatelské rozhraní

Grafické uživatelské rozhraní bylo navrženo s ohledem na jednoduché ovládání a pohodlí uživatele. Návrh byl proveden pomocí nástroje [www.draw.io](http://www.draw.io). Návrh čtyř nejvíce důležitých a nejsložitějších oken je uveden níže.

- **Základní okno**

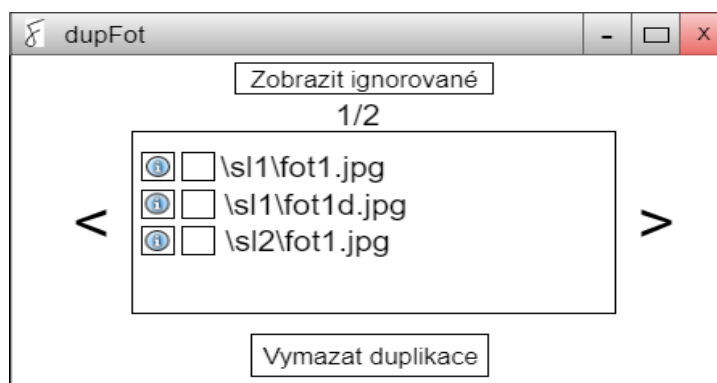
Musí uživateli umožnit zadání zdrojových složek, nastavení parametrů vyhledání duplicit a také jeho spuštění.



Obrázek 4.3: Náčrt základního okna aplikace

- **Okno se skupinami duplicit**

Zobrazí vyhledané duplicitní skupiny. Na základě zdrojů a nastavení. Musí uživateli umožnit manuální výběr, ale zároveň zobrazit automatický výběr aplikace. Také zde musí být možnost zobrazit ignorované fotografie, které neobsahují Exif metadata.

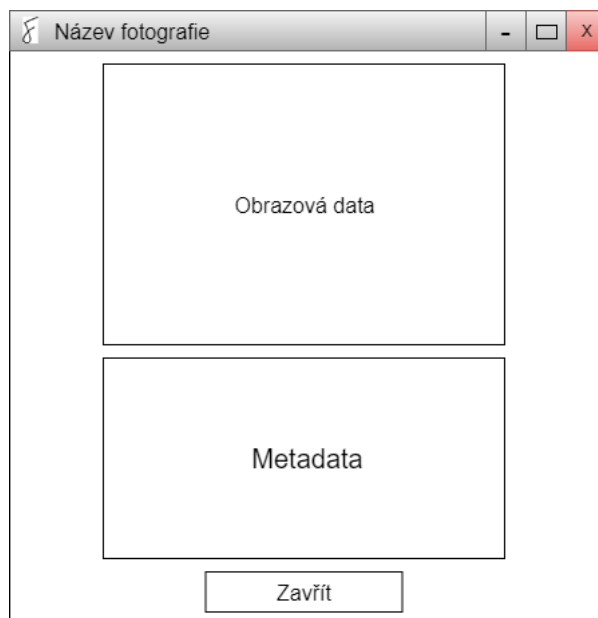


Obrázek 4.4: Náčrt okna duplicit aplikace

Některé funkce jako třeba export fotografií nebo editace metadat byly do aplikace přidány až v samotném závěru vývoje a v původním návrhu tak jejich ovládací prvky chybí.

- **Okno s informacemi o fotografii**

Musí uživateli zobrazit všechny dostupné informace o fotografii. Pro usnadnění identifikace fotografie mezi ostatními. Uživatel si tedy po zobrazení tohoto okna musí být jistý o jakou fotografii se jedná.



Obrázek 4.5: Náčrt okna s informacemi o fotografii

- **Okno nastavení**

Musí uživateli umožnit změnu a zobrazení aktuálního nastavení. Také návrat k původnímu nastavení aplikace.

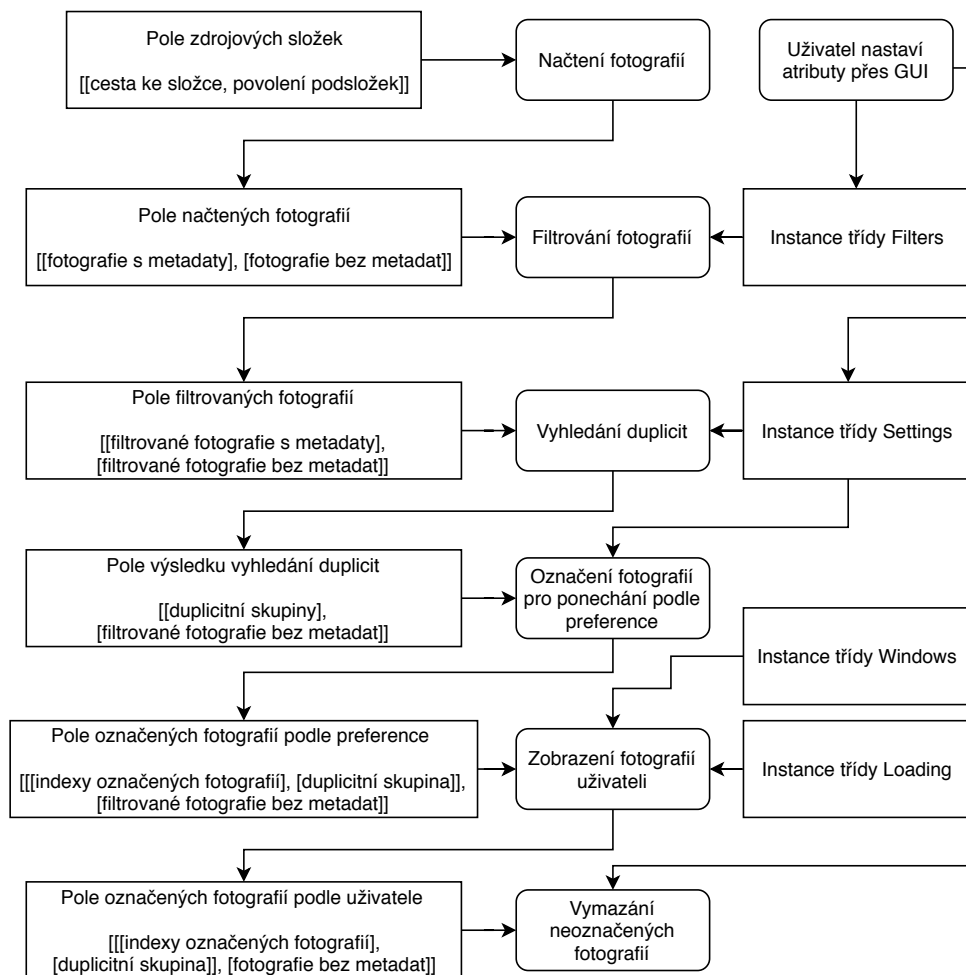


Obrázek 4.6: Náčrt okna s nastavením aplikace

## Kapitola 5

# Implementace

Po definování všech potřebných funkcí v předchozí části vývoje. Bylo potřeba vypracovat reálné řešení zadaných specifikací ve zvoleném implementačním prostředí.



Obrázek 5.1: Schéma modulů aplikace

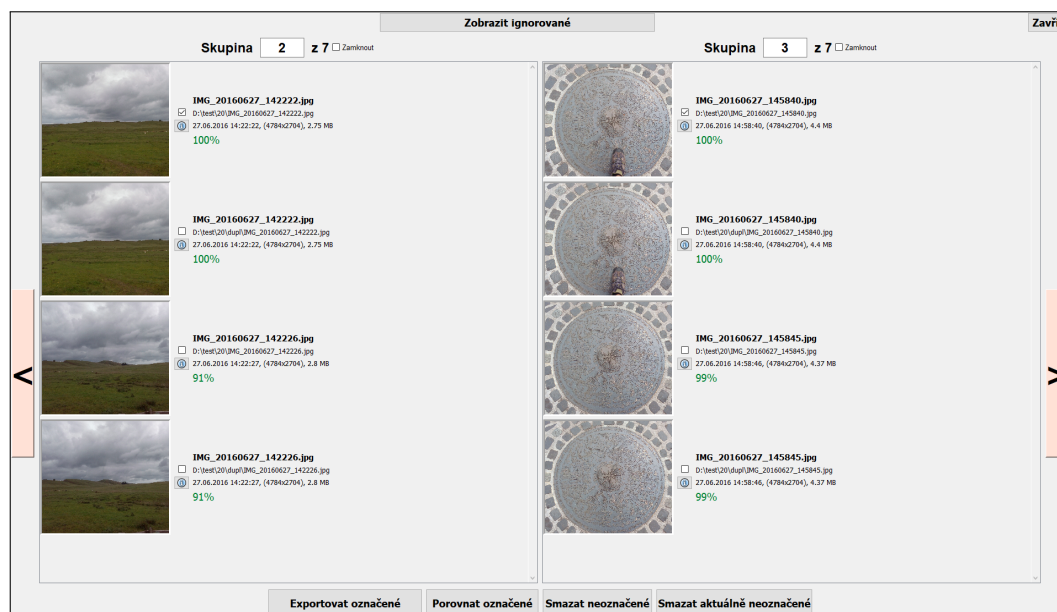
Fotografie jsou uloženy v poli jako instance třídy `Image`. Vyhledání duplicit je podrobněji popsáno samostatně v kapitole 5.2.3. Pole je v jazyce Python3 alokované dynamicky.

Nejprve byly implementovány všechny funkce aplikace tedy tzv. backend, který byl vyvíjen a testován po částech. Každá část tedy byla nejprve implementována a otestována samostatně. Tímto byla zajištěna správná funkčnost každého modulu, každý z nich totiž měl předem jasné požadavky, které musel pro splnění svého účelu v aplikaci splnit. Poté byly tyto části propojeny až vznikl kompletní backend aplikace.

Na backend aplikace byl postupně napojován frontend. Vzhledem k tomu, že jsem se knihovnu `pyQT5` musel postupně naučit vznikl nejdříve pouze v testovací verzi bez lepšího designu. Teprve poté, když byly napojeny a otestovány všechny funkce backendu přes frontend, začal vývoj frontendu, který nyní aplikace má.

## 5.1 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní je implementováno použitím knihovny `PyQt5` v souboru `gui.py`. Každá grafická komponenta má zde svoji třídu, která dědí od třídy `QWidget` a také funkci pro její správné zobrazení a uzavření. Všechny aktivní grafické komponenty jsou uloženy do společného seznamu, mohou tak být hromadně uzavřeny. Zároveň je nutné uchovávat referenci na instanci kvůli `garbage collectoru`. Vzhled většiny oken je realizován pomocí layoutů, které umožní změnu velikosti okna a libovolné přidávání nových prvků. Při zachování jejich umístění. Některá okna umožňují použití klávesových zkratk.

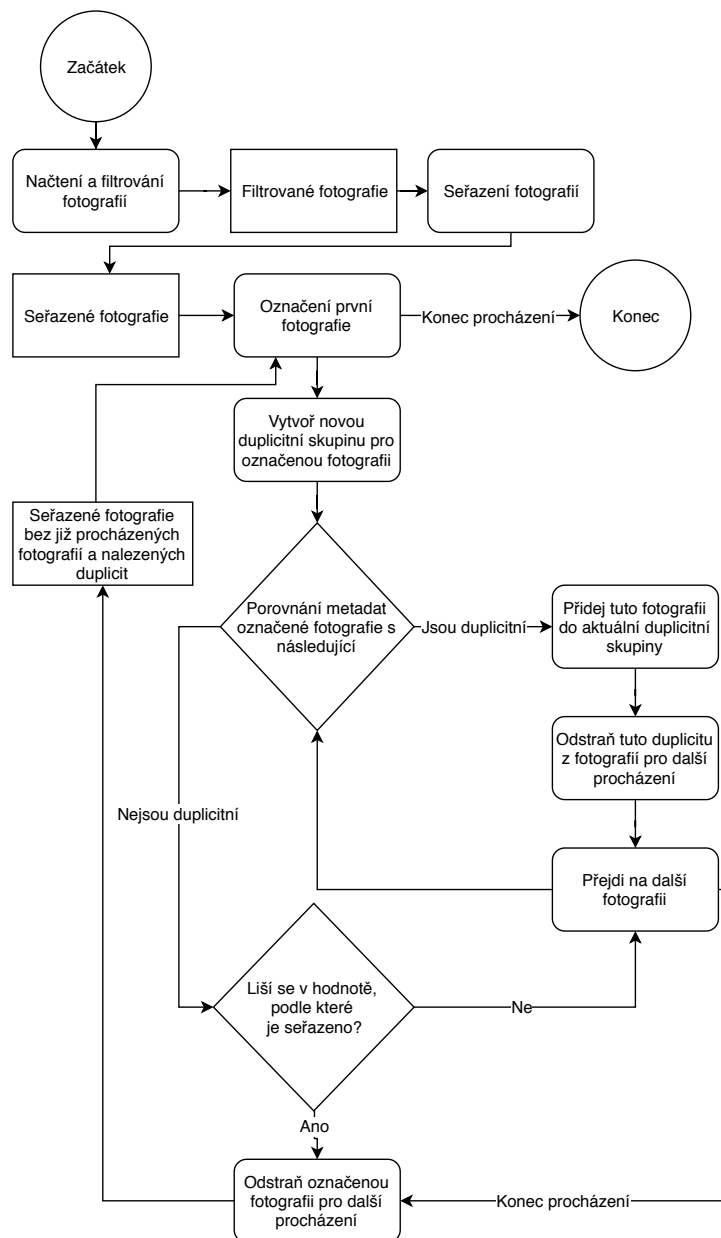


Obrázek 5.2: Ukázka zobrazení duplicitních skupin

Nejdůležitějším oknem celé aplikace je okno pro zobrazení výsledků vyhledávání uživateli. Každá fotografie zde má náhled obrazu, svůj název, celou cestu k ní, datum pořízení, rozlišení, velikost, percentil lokální kvality ve skupině (pokud je zapnuta). Také je zde možnost každou fotografii označit pro ponechání či export nebo porovnání s ostatními. Také lze u každé fotografie zobrazit její detail, kde jsou zobrazena obrazová data s možností zoomu a její metadata. V tomto okně se lze pohybovat klávesovými nebo grafickými šipkami. Lze přejít rovnou na jakoukoliv skupinu z nalezených. Také je možné zamknout skupinu a porovnat ji tak postupně se všemi ostatními.

## 5.2 Vyhledání duplicit

Hlavní funkcí aplikace je vyhledání duplicit, tento algoritmus byl rozdělen do několika částí. Tato část je uvedena obecně v kapitole 4.3.1 jako Vyhledání duplicit ve vyfiltrovaných fotografiích podle nastavených parametrů. Má již tedy nastavené parametry a zdroje od uživatele.



Obrázek 5.3: Obecný vývojový diagram vyhledávání duplicit v načtených fotografiích

Podmínky pro porovnání metadat jsou blíže popsány v kapitole 5.2.3. Pro seřazení fotografií je využit TimSort, který je vestavěným řadícím algoritmem v jazyce Python3 [20] a má složitost:  $O(n * \log(n))$ .

### 5.2.1 Vstupní nastavení

Mezi hlavní parametry, které lze nastavit patří:

- způsob mazání fotografií (koš, úplné)
- preferenci automatického výběru (nic, velikost, datum poslední změny, datum pořízení, priorita složky, rozlišení, percentil kvality, každou možnost lze znegovat)
- v případě shody preference u více fotografií lze označit pouze jednu nebo všechny s odpovídající hodnotou
- lze vyloučit z porovnávání záznam o rozlišení v metadatech, platné pouze při hledání úplné duplicity
- hledání úplné nebo částečné duplicity metadat
- zobrazení pouze skupin duplicit nebo i jednotlivých fotografií
- prohledávání podsložek pro nově zadané složky
- velikost tolerovaného časového intervalu při porovnávání datumů
- vyhodnocení duplicity také pouze na základě stejného názvu souboru (pokud mají dvě fotografie v jiných složkách stejný název tak jsou duplicitní)
- zapnutí a vypnutí vyhodnocení kvality
- zvolení vlastností které mají být zahrnuty při vyhodnocení kvality (Ostrost, Šum, Barevnost, čím více položek tím větší čas trvá kvalitu vyhodnotit)
- zvolení počtu zobrazených duplicitních skupin
- výběr podle čeho budou skupiny seřazeny (čas, počet fotografií ve skupině)
- hledání duplicity (částečná nebo úplná duplicita metadat, vyloučení rozlišení z metadat)
- zobrazení metadat při ručním paralelním porovnání fotografií

Nastavení se ukládá do souboru při jeho první změně a potvrzení. Je vytvořen soubor `dupFot.sett`. Který obsahuje uloženou instanci třídy `Settings`. Při spuštění aplikace je primárně načítán tento soubor, pokud neexistuje jsou nastaveny výchozí hodnoty. Aplikace obsahuje výchozí nastavení, které uživatel může obnovit resetem svých hodnot nebo smazáním souboru s uloženým nastavením.

### 5.2.2 Načtení fotografií

Uživatel zadá zdrojové složky a u každé nastaví, zda mají při hledání být zahrnuty i podsložky. Úroveň zanoření potom už není nijak omezena. Jako výchozí adresář pro selekci je zvolena nadřazená složka poslední zadané složky. Pokud ji nemá, tak je to poslední vybraná složka. Pro první výběr je nastavena výchozí složka na `\home`.

Pro každou zdrojovou složku je vytvořeno jedno vlákno, to pak projde všechny podsložky a soubory samostatně. Vlákna se synchronizují na konci procházení. Načteny jsou všechny fotografie, které se nacházejí v zadaných složkách, popřípadě podsložkách. Primárně jsou vyhledány Exif metadata, sekundárně jsou pak hledány IPTC údaje o fotografii. Pokud fotografie nemá ani jeden typ z těchto metadat je zahrnuta do seznamu ignorovaných. Výstupem načtení jsou tedy dvě skupiny fotografií.

### 5.2.3 Vyhledání duplicity podle metadat

V načtených fotografiích, které obsahují metadata jsou vyhledány duplicity. Všechny fotografie jsou načteny do jednoho pole. Toto pole je seřazeno podle datumu pořízení fotografie, případně podle typu fotoaparátu nebo výrobce. Záleží na tom, který záznam je přítomen u všech fotografií. V naprosté většině případů je pole seřazeno podle datumu pořízení fotografie. Poté je pole postupně procházeno. Každá fotografie je porovnána s následujícími od své pozice. Vyhledávání jde tedy pouze dopředu. Nalezené duplicity jsou vyloučeny z dalšího porovnávání. Pro každou fotografii tedy platí, že bude mít minimálně o jedna méně porovnání než předchozí. Také platí, že porovnávací algoritmus se může zastavit pro aktuální fotografii v momentě kdy nalezne neshodu v záznamu, podle kterého je pole seřazeno. Může tedy ukončit hledání pro aktuální fotografii a přejít na další. Ignorované fotografie nejsou porovnávány.

Dále jde nastavit, zda se má jednat o úplnou, nebo částečnou duplicitu. Při částečné shodě jsou porovnány pouze údaje: datum pořízení fotografie, výrobce fotoaparátu a jeho typ/model. Je tedy pravděpodobné, že hledání úplné duplicity metadat najde pouze kopie, nikoliv upravené fotografie. Upravenou fotografií je zde myšlena modifikace obrazových dat, metadata by se v takovém případě měla přenést, nicméně editory mají tendenci přidávat nebo měnit hodnoty některých záznamů. Porovnávání fotografií končí se záporným výsledkem v případě nalezení první neshody v hodnotách záznamů metadat. Při porovnávání datumu pořízení fotografie je zohledněn tolerovaný časový interval, který je zadán uživatelem.

Duplicitní fotografie jsou seskupeny a je proveden automatický výběr nejvhodnější pro ponechání podle uživatelem nastavené preference. Uživatel jej může úplně vypnout, popřípadě zvolit jiné kritérium z nabízených možností. Na konci každého vyhledání duplicit jsou uživateli zobrazeny jeho statistiky.

V průběhu načtení fotografií a vyhledání duplicit je uživateli zobrazen odhadovaný čas dokončení operace. Odhadovaný čas je vypočítán jako rozdíl aktuálního času a počátečního času načítání. Tento rozdíl je vydělen počtem hotových procent. Tím je získán průměrný čas na jedno procento. Tento průměrný čas je vynásoben počtem zbývajících procent. Tedy rozdílem 100 a počtem hotových procent. Tento čas se mění každé další procento. Je tedy čím dál více přesnější a zmenšuje se.



## 5.3 Vyhodnocení kvality fotografie

V nalezených skupinách duplicitních fotografií je potřeba podle jednotného měřítka určit fotografii, kterou si uživatel chce ponechat. Aplikace tedy má možnost automatické preference podle záznamů z metadat a poté vyhodnocení kvality podle obrazových dat. Popřípadě obsahuje nástroje pro ruční porovnání obrazových dat zvolených fotografií.

### 5.3.1 Automatické

Po sestavení duplicitních skupin je pro každou fotografii vypočteno ohodnocení vlastností fotografie. Je vyhodnocena míra šumu, barevnost a ostrost fotografie. Všechny tyto položky jsou volitelné a uživatel si může zvolit, které budou vyhodnoceny. Hodnoty jsou vypočteny kombinací matematických operací a transformacemi obrazu. Tyto hodnoty jsou sečteny do sumy. Percentil kvality je vyhodnocen vždy v rámci duplicitní skupiny. Fotografie s největší sumou hodnot je tedy ohodnocena jako 100 % a její suma je stanovena jako maximum skupiny. Ostatní fotografie jsou ohodnoceny na základě poměru své sumy k maximum skupiny.

### 5.3.2 Manuální

Automatické vyhodnocení kvality není a ani nemůže být vždy přesné. Proto jsou zde možnosti pro uživatelské porovnání fotografií. Uživatel může vybrat fotografie do společného porovnání obrazu. V tomto porovnání je umožněn provázaný zoom fotografií.

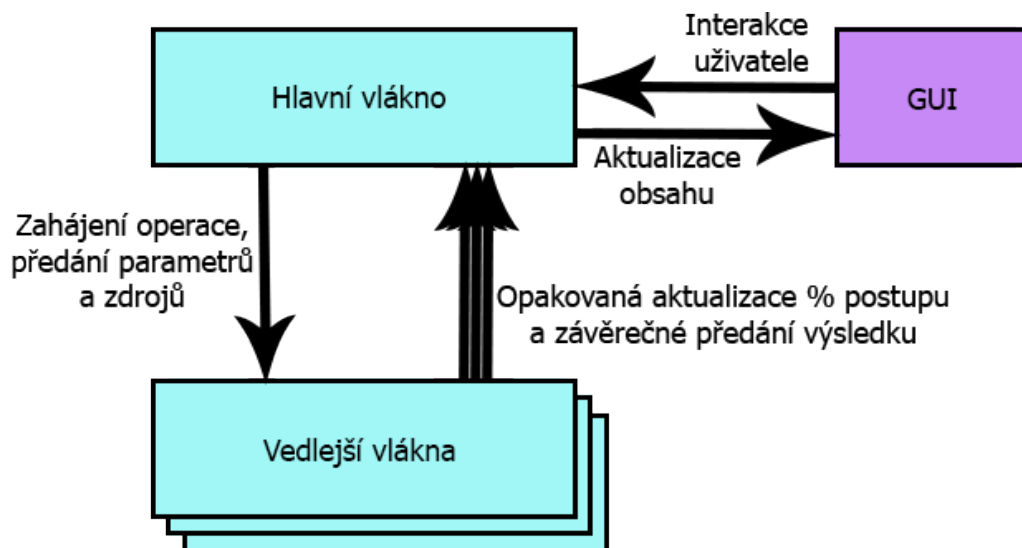


Obrázek 5.4: Ukázka manuálního porovnání duplicit

V tomto porovnání lze označit na jedné fotografii oblast kterou chcete porovnat, tato oblast se zobrazí zvětšená u všech fotografií. Stejně tak je možné dále provázaně postupně přibližovat tuto oblast. Jediná akce, která není propojená je posunutí. Z důvodu menších rozdílů rozměrů se tedy lze posunout na správné místo ve fotografii ručně. Provázané přibližování funguje při aspoň přibližně stejných rozměrech porovnávaných fotografií.

## 5.4 Realizace vláken

Výsledkem vývoje je více vláknová aplikace se signály pro synchronizaci. Hlavní vlákno obsluhuje GUI, na pozadí probíhají ostatní procesy, které mohou vytvořit další vlákna. Hlavní vlákno je synchronizováno pomocí signálů. Pokud je úloha v pozadí dokončena je emitován příslušný signál a GUI je aktualizováno. Vlákna jsou vytvořena za pomoci vestavěné knihovny `threads`.



Obrázek 5.5: Náhled vláken aplikace

Pokud je v průběhu vykonávání aplikace vyvolána neošetřená výjimka, je zachycena a její popis je zapsán do logovacího souboru. Soubor se vytvoří automaticky při prvním spuštění aplikace a poté už jenom navazuje na předchozí. Nemaže tedy žádnou chybu. Tento soubor má nastavený pevný název `dupFot.log`. Uzavření hlavního okna ukončí celou aplikaci.

V této kapitole byl popsán způsob realizace návrhu aplikace z předcházející kapitoly 4, výsledek implementace je vyhodnocen v následující kapitole.

## Kapitola 6

# Vyhodnocení aplikace

V průběhu vývoje aplikace je nutné ji průběžně testovat. Příklady nejčastějších testů:

- uživatelské testy, otestování GUI a funkcí aplikace
- výkonnostní a verifikační testy kvůli ověření správného směru vývoje
- zátěžové testy kvůli ověření stability aplikace
- využití prostředků kvůli stanovení minimálních požadavků a profilování aplikace
- porovnání s konkurencí na základě jednotného testu kvůli zjištění čím je aplikace jiná a čím se tedy prosadí v používání mezi uživateli

V případě mojí aplikace bylo nutné průběžně vyhodnotit: rychlost, správnost a stabilitu aplikace. Pro tento účel vznikla testovací galerie, která měla předem dané výsledky, kterých má aplikace dosáhnout. Také musel být jednoznačně určen testovací stroj kvůli porovnávání výkonu. Časem byl tento test automatizován a rozšířen tak, aby bylo možné pomocí něj testovat i stabilitu aplikace. Pro porovnání s konkurencí byl použit pro všechny aplikace stejný test. Který měl za úkol odhalit hlavně časovou náročnost vyhledání duplicit.

## Parametry testovacího stroje

- Operační systém: Windows10 Education, 16299.371, 64bit (na SSD)
- Základní deska: Gigabyte 970A-D3
- SSD: KINGSTON SHSS37A240G
- Procesor: AMD FX-4350, Quad-Core 4.20 GHz
- RAM: 24 GB, DDR3, 2133 MHz, CL11
- HDD: WDC WD20EZR-00D8PB0, 2TB, 7200 RPM, cache 64 MB
- GPU: AMD Radeon R9 200 Series, 3072 MB, GDDR5, 1100 MHz

Jako HDD je uveden disk, na kterém byla umístěna aplikace a testovací galerie. Všechny testy proběhly v minimálním zatížení stroje ostatními aplikacemi či procesy.

## Testovací galerie

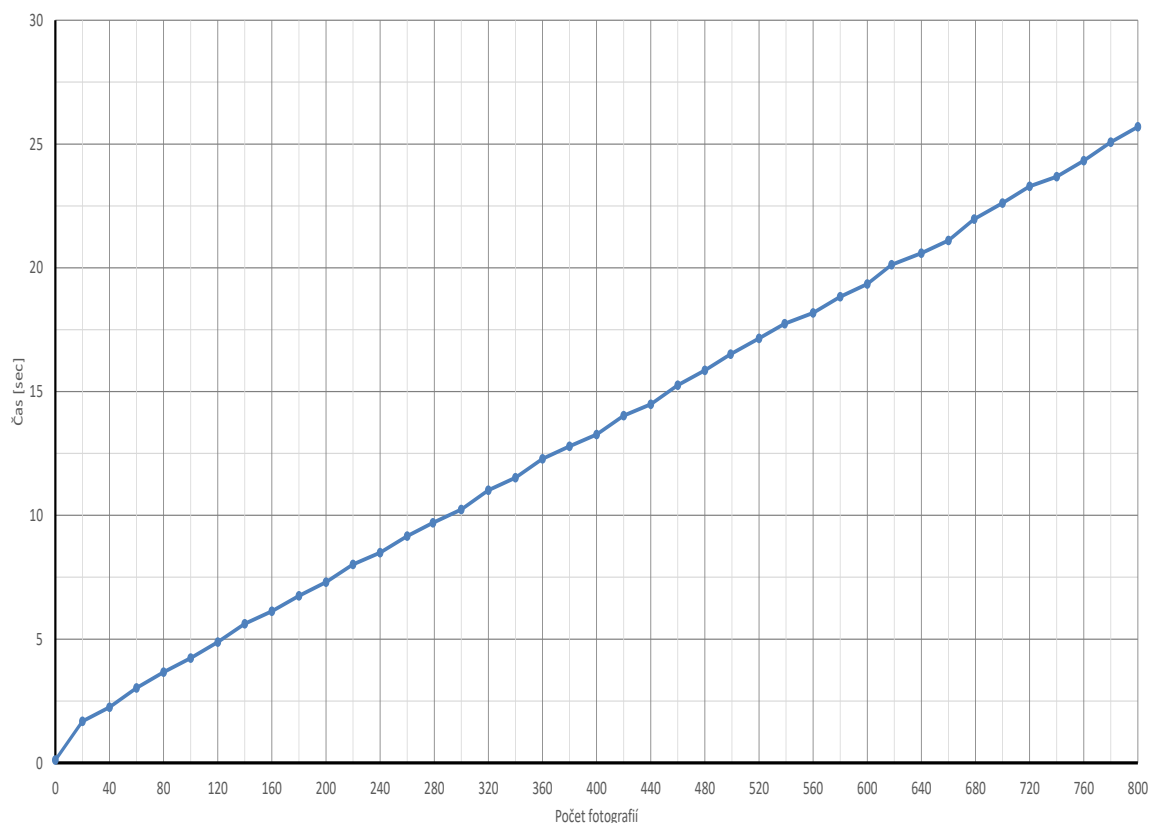
Skládá se z 40 složek. Každý adresář vždy obsahuje 10 fotografií a jejich duplicity umístěné ve svém podadresáři. Duplicity mohou být přesné kopie ale i upravené fotografie. Testované editory jsou Adobe Photoshop CS6 a Gimp. V testovací galerii jsou pouze fotografie formátu JPG. Celkově je tedy zde 400 unikátních a 400 duplicitních fotografií. Průměrná velikost fotografie je 2,6125 MB.

Galerie zde simuluje jednu ze složek dlouholeté galerie. Tedy například běžnou jedno-týdenní dovolenou. Kde uživatel pořídí určité množství fotografií, po návratu z ní uživatel některé fotografie následně upravuje například: redukuje šum, otáčí, celé vloží duplicitně, změni velikost větší komprimací, udělá výřez. Některé fotografie jsou časem pořízeny velice blízko a je nepravděpodobné, že by uživatel za tento čas stihl udělat další jedinečnou fotografii, velikost tohoto intervalu lze nastavit. Všechny tyto duplicity aplikace detekuje a seskupí.

### 6.1 Automatické testování

Automatické testování je důležité pro včasné odhalení chyby v nové verzi aplikace. Díky tomu že je automatické může být spuštěno i na dálku na referenčním stroji po přenesení aktuální verze aplikace. Také může být v rychlém sledu za sebou spuštěno několik testů. Výsledky testu jsou po provedení zapsány a uchovány pro pozdější porovnání s další verzí. Tímto je zajištěno, že každá nová verze je porovnána s tou předchozí. Každé otestování tedy prověří: rychlost, stabilitu a správnost nové verze aplikace.

Pro automatické, výkonnostní testování byl vytvořen skript `test.py`. Který každé vyhledání provede desetkrát, zaznamená dobu trvání každého vyhledání a zapíše medián z těchto výsledných časů. Poté přidá další složku do zdrojů a pokračuje. První průběh testování probíhá bez podsložek. Všechny fotografie jsou tedy jedinečné. **Druhý průběh** testování zahrne i podsložky, to tedy znamená mnohem více fotografií, ale také mnoho duplicit. Všechny výsledky jsou automaticky zapsány do `xlsx` souborů. Cílem je ověřit stabilitu a získat rychle výkonnostní údaje celé aplikace. Také jej lze použít pro rychlé ověření správnosti nalezeného počtu fotografií a jejich duplicit v zadaných složkách. Pro tento účel je nutné snížit počet opakování tedy konkrétně proměnnou `count` na 1.



Obrázek 6.1: Graf výsledků 2.běhu automatických testů

Podrobné údaje ze všech měření lze najít ve zdrojových souborech dokumentace ve složce `tests`. Na grafu můžeme sledovat lineární nárůst potřebné doby zpracování na více fotografií. Měření bylo spuštěno s výchozím nastavením, nad aplikací prováděné interpretem, která tedy nebyla nijak zkompileována.

## 6.2 Porovnání s konkurencí

Pro objektivní hodnocení aplikace je také důležité porovnání s konkurencí. Aplikace by měla nabídnout něco navíc a také se něčím odlišit. Do porovnání s mojí aplikací **dupFot** jsou zahrnuty vlastnosti: vzhled, dostupnost, způsob určení duplicity a jeho účinnost.

Všem aplikacím bylo zadáno vyhledat duplicity z celé testovací galerie. U této operace byly zkoumány i ostatní aspekty aplikací. Všechny aplikace jsou blíže popsány v kapitole 3.

Název aplikace	GUI	Určení duplicity	Licence	Správně	Čas [min]
dupFot kompil.	Ano	Metadata	Otevřená	Ano	0.24
dupFot interp.					0.42
VS DIF Demo	Ano	Metadata	Omezená	Přijatelné	1.23
		Obraz			1.57
DP Fixer Pro	Ano	Obraz	Omezená	Ano	2.55
VisiPics	Ano	Obraz	Zdarma	Ne	X
ExifTool	Ne	Metadata	Zdarma	Nelze	X
Anti-Twin	Ano	Bitové porovnání	Zdarma	Ano	25.16
		Obraz			146.78

Tabulka 6.1: Porovnání aplikace dupFot s konkurencí

DP Fixer - Duplicate Photos Fixer

VS DIF - Visual Similarity Duplicate Images Finder

dupFot kompil. - zkompilevaná aplikace dupFot pomocí knihovny cx\_Freeze [28]

dupFot interp. - viz. graf 6.1

Výsledek byl správně, pokud aplikace našla všechny duplicity v testovací galerii pouze zadáním nejvyšší zdrojové složky. Přijatelný výsledek znamená sloučení některých hodně podobných skupin. Demo aplikace VS DIF nezobrazí všechny skupiny duplicit.

## 6.3 Možná rozšíření aplikace

Jako možná rozšíření aplikace do budoucna vyplynula během analýzy a konzultací s vedoucím práce. Jako rozšíření tedy budou zvažovány funkce: hromadná editace metadat, export výběru fotografií omezený časovými intervaly a bez duplicit, vytvoření databáze už jednou zpracovávaných fotografií, určení duplicity podle obrazových dat za použití již implementovaných metrik s procentuální shodou a tresholdem, určení duplicity podle porovnání byte-to-byte, klávesové zkratky pro ovládání GUI.

## 6.4 Shrnutí výsledků

Moje aplikace prošla automatickými testy s přijatelnou rychlostí a správnými výsledky. Interpretem spuštěná aplikace předčila svojí rychlostí ověřované aplikace. Zkompilevaná aplikace splnila předpoklady a je ještě zhruba o polovinu rychlejší. Díky porovnávání časové shody metadat a časovému intervalu místo přesného času je možné efektivně seskupit duplicity. Nicméně v případě implementace vytvoření databáze již zpracovaných fotografií by mohla aplikace dosáhnout ještě větší rychlosti.

## Kapitola 7

# Závěr

Cílem této bakalářské práce bylo zejména vytvořit aplikaci, která vyhledá duplicity ve skupině fotografií podle Exif metadat a umožní uživateli jejich efektivní porovnání a následně hromadné vymazání či přesun zvolených fotografií. Tento cíl byl splněn.

V průběhu práce jsem nastudoval literaturu se zaměřením na téma informací obsažených v digitálních fotografiích. Dále jsem vyhledal existující řešení pro vyhledání duplicit ve fotografiích. S prioritou pro vyhledávání podle metadat. Zjištěné vlastnosti existujících aplikací a nastudovanou teorii se zadáním práce jsem využil pro specifikaci požadavků na aplikaci a navrhnutí postupu vyhledávání duplicit ve fotografiích podle doplňkových informací. Na základě požadavků jsem vyhledal vhodná implementační prostředí a zvolil Python ve verzi 3.6. Poté jsem navrhl a implementoval jednotlivé položky grafického rozhraní s použitím knihovny `pyQT5`. Výslednou aplikaci jsem otestoval na základě cíle a požadavků a porovnal s nalezenými aplikacemi.

Aplikace svými funkcemi splnila zadání, a také přidává několik dalších. Obstála v automatických testech, kde prokázala svoji rychlost a stabilitu. Dále je plně vyřešena distribuce aplikace k uživatelům. Při jakékoliv úpravě stačí nainstalovat pouze interpret pro jazyk Python3 a spustit kompilační skript, potřebné knihovny jsou staženy automaticky. Tento skript automaticky vytvoří novou verzi aplikace včetně spustitelného souboru. Nastavení aplikace lze přenášet spolu s aplikací. Zkompilovaná aplikace tedy může být použita i jako přenosná aplikace.

S volbou programovacího jazyka jsem spokojen, splnil vše, co se od něj očekávalo a v průběhu vývoje nebyl důvod k přehodnocení této volby. Vývoj složitější aplikace v jazyce Python 3.6.2 byl velmi zajímavý a v mnoha ohledech jiný než u jeho konkurence. Nicméně nakonec se mi vždy podařilo dosáhnout toho co jsem potřeboval.

V budoucím vývoji aplikace je potřeba se zaměřit hlavně na otestování dalších platforem. Dále se zaměřit na rozšíření možností hromadné editace metadat. Přidání dalších metrik pro vyhodnocení kvality. Také zavedení vytváření databázového souboru již zpracovávaných fotografií, pro ještě rychlejší načítání informací.

# Literatura

- [1] XMP adding intelligence to media. Adobe Systems Incorporated, San Jose (CA), 2012, [Online; navštíveno 20.07.2018].  
URL <https://www.adobe.com/products/xmp.html>
- [2] *.ZIP File Format Specification*. PKWARE Inc., 6 vydání, 10 2014, [Online; navštíveno 02.07.2018].  
URL <https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT>
- [3] Visual Similarity Duplicate Images Finder. MindGems Inc., Langley, 2015, [Online; navštíveno 12.06.2018].  
URL <http://www.mindgems.com/products/VS-Duplicate-Image-Finder>
- [4] Top 5 Best Duplicate Photo Finder Tools to Remove Duplicate Photos. *ashisoft.com-blog*, 7 2017, [Online; navštíveno 13.07.2018].  
URL <https://www.ashisoft.com/blog/top-5-best-duplicate-photo-finder-to-delete-duplicate-photos>
- [5] DuplicatePhotosFixer. Systweak Software, Jaipur, 2018, [Online; navštíveno 12.06.2018].  
URL <https://www.duplicatephotosfixer.com>
- [6] Photo Metadata. International Press Telecommunications Council, 2018, [Online; navštíveno 20.07.2018].  
URL <https://iptc.org/standards/photo-metadata>
- [7] The Python Tutorial. *Python 3.6.2 documentation*, 07 2018, [Online; navštíveno 16.07.2018].  
URL <https://docs.python.org/3/tutorial>
- [8] TIOBE Index for Python. TIOBE software BV, Eindhoven, 07 2018, [Online; navštíveno 26.07.2018].  
URL <https://www.tiobe.com/tiobe-index/python>
- [9] Carbone, P.: PYPL PopularitY of Programming Language. 07 2018, [Online; navštíveno 26.07.2018].  
URL <http://pypl.github.io/PYPL>
- [10] David Hasler, S. S.: *Measuring Colourfulness in Natural Images*. Lausanne: Swiss Fed. Inst. of Tech., 2018, [Online; navštíveno 21.07.2018].  
URL <https://infoscience.epfl.ch/record/33994/files/HaslerS03.pdf>



- [11] Fouet, G.: VisiPics. 07 2008, [Online; navštíveno 12.07.2018].  
URL <http://www.visipics.info>
- [12] Harvey, P.: ExifTool. Cambridge, 07 2018, [Online; navštíveno 12.07.2018].  
URL <https://www.sno.phy.queensu.ca/~phil/exiftool>
- [13] Hegmon, J.: *Automatická anotace obrazu*. Diplomová práce, Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, Brno, 2013.  
URL <http://hdl.handle.net/11012/26552>
- [14] Hrastnik, B.: ExifToolGUI. 05 2012, [Online; navštíveno 12.06.2018].  
URL <http://u88.n24.queensu.ca/~bogdan>
- [15] Jahoda, B.: Jaký formát obrázku použít na webu. *Je čas-Obrázky*, 03 2015, [Online; navštíveno 20.07.2018].  
URL <http://jecas.cz/format-obrazku>
- [16] Khatri, M.: 5 Best Duplicate Photo Finder Tools to Delete Duplicate Photos. *Top 10*, 04 2016, [Online; navštíveno 13.07.2018].  
URL <https://blogs.systweak.com/2016/04/5-duplicate-photo-finder-tools-to-delete-duplicate-photos>
- [17] Král, B.: *Analýza a identifikace šumu fotoaparátu*. Bakalářská práce, Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav informačních systémů, Brno, 2015.  
URL <http://hdl.handle.net/11012/52354>
- [18] Kužel, J.: *Editor EXIF a IPTC údajů z fotografií*. Bakalářská práce, Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav inteligentních systémů, Brno, 2011.  
URL <http://hdl.handle.net/11012/55635>
- [19] Macenauer, A.: Co je digitální fotografie. *FotoRádce*, 03 2002, [Online; navštíveno 20.07.2018].  
URL <https://www.fotoaparar.cz/clanek/230>
- [20] Peters, T.: Timsort. 2002, [Online; navštíveno 26.06.2018].  
URL <https://svn.python.org/projects/python/trunk/Objects/listsort.txt>
- [21] Peters, T.: The Zen of Python. *Python Developer's Guide*, 8 2004, [Online; navštíveno 16.07.2018].  
URL <https://www.python.org/dev/peps/pep-0020>
- [22] Petr Lindner, T. T., Miroslav Myška: *Velká kniha digitální fotografie*. Brno: CP Books, druhé vydání, 2005, ISBN 80-251-0648-9.
- [23] Pihan, R.: Formáty pro ukládání fotografií - 7.díl: TIFF. *DigiManie*, 12 2007, [Online; navštíveno 16.07.2018].  
URL <https://www.digimanie.cz/formaty-pro-ukladani-fotografii-7dil-tiff/2023>

- [24] Rosenthal, J.: Anti-Twin. Lippstadt, 05 2010, [Online; navštíveno 12.06.2018].  
URL <http://www.joerg-rosenthal.com/en/antitwin>
- [25] Sokolov, V.: Handling ZIP Archives in pure MQL5. *MQL5 Programming Articles*, 10 2015, [Online; navštíveno 16.07.2018].  
URL <https://www.mql5.com/en/articles/1971>
- [26] Soukup, R.: *Škola digitální fotografie*. Praha: Grada Publishing a.s., první vydání, 2006, ISBN 8024710773.
- [27] Tim Park, D. D.: Identifying duplicate electronic content based on metadata. 09 2012.  
URL <https://patents.google.com/patent/US8266115B1/en>
- [28] Tuininga, A.: *cx\_Freeze Documentation*. Edmonton: Computronix (Canada) Ltd., 6 vydání, 12 2017, [Online; navštíveno 25.06.2018].  
URL <https://media.readthedocs.org/pdf/cx-freeze/latest/cx-freeze.pdf>
- [29] Večeřa, V.: *Obnova hesel archivů ZIP s využitím GPU*. Bakalářská práce, Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav informačních systémů, Brno, 2016.  
URL <http://hdl.handle.net/11012/62038>
- [30] Wikipedia contributors: Exchangeable image file format. *Wikipedia: the free encyclopedia*, 04 2018, [Online; navštíveno 16.07.2018].  
URL <https://cs.wikipedia.org/wiki/Exif>
- [31] Wikipedia contributors: Graphics Interchange Format. *Wikipedia: the free encyclopedia*, 07 2018, [Online; navštíveno 17.07.2018].  
URL <https://cs.wikipedia.org/wiki/GIF>
- [32] Wikipedia contributors: Joint Photographic Experts Group. *Wikipedia: the free encyclopedia*, 05 2018, [Online; navštíveno 14.07.2018].  
URL <https://cs.wikipedia.org/wiki/JPEG>
- [33] Wikipedia contributors: Portable Network Graphics. *Wikipedia: the free encyclopedia*, 07 2018, [Online; navštíveno 15.07.2018].  
URL [https://cs.wikipedia.org/wiki/Portable\\_Network\\_Graphics](https://cs.wikipedia.org/wiki/Portable_Network_Graphics)
- [34] Wikipedia contributors: RAW (grafika). *Wikipedia: the free encyclopedia*, 04 2018, [Online; navštíveno 18.07.2018].  
URL [https://cs.wikipedia.org/wiki/Raw\\_\(grafika\)](https://cs.wikipedia.org/wiki/Raw_(grafika))
- [35] Wikipedia contributors: Tagged Image File Format. *Wikipedia: the free encyclopedia*, 06 2018, [Online; navštíveno 16.07.2018].  
URL [https://cs.wikipedia.org/wiki/Tagged\\_Image\\_File\\_Format](https://cs.wikipedia.org/wiki/Tagged_Image_File_Format)
- [36] Černoch, P.: Digitální fotografie - vznik, historie, současnost a vývojové trendy. *Historie a vývojové trendy ve výpočetní technice*, 05 2003, [Online; navštíveno 20.07.2018].  
URL <https://www.fi.muni.cz/usr/jkucera/pv109/2003/xcernoc1>

# Příloha A

## Obsah přiloženého média

Přiložené médium obsahuje repozitář aplikace dupFot 2.1, aktuální k datumu 31.07.2018.

- **zdrojový kód**  
Včetně komentářů ke každé třídě, metodě, funkci a souboru samotnému na jeho na začátku.
- **dokumentace**  
Vytvořené PDF a všechny zdrojové soubory  $\text{\LaTeX}$  včetně Makefile a použitých obrázků.
- **zkompilovaná aplikace**  
Pro OS typu Windows. Na ostatních OS nebylo provedeno testování ani kompilace, nicméně kompilovací skript by měl být připraven na OS typu Linux a MacOS.

## Rozdělení zdrojového kódu

Kvůli lepší orientaci ve zdrojovém kódu aplikace. Byla zavedena následující hierarchie. Každý soubor tvoří samostatný celek a využívá předchozích souborů ve směru odshora dolů v následujícím pořadí. Více informací lze naléznout v komentářích v každém z nich.

- **classes.py**  
Soubor pro třídy, které se netýkají GUI.
- **globalVars.py**  
Soubor pro vytvoření globálních proměnných.
- **lib.py**  
Soubor pro funkce backendu aplikace.
- **gui.py**  
Soubor pro třídy a jejich metody, funkce pro vytvoření a práci s GUI.
- **main.py**  
Soubor pro spuštění aplikace.

- **setup.py**  
Soubor pro kompilaci aplikace.
- **test.py**  
Soubor pro testování aplikace. Může také sloužit jako ukázka pro další použití v jiných programech. Není potřeba pro kompilaci ani běh aplikace. viz kapitolu [6](#).

## Instalace aplikace

Je provedena spuštěním instalačního skriptu. Který je naprogramován jako vše v jazyce Python. Pro jeho spuštění je nutné zadat v příkazové řádce/terminálu následující příkaz:  
`~/dupFot/app>python ./setup.py build`

Pro kompilaci aplikace je tedy nutné mít nainstalovaný Python 3.6.x. Potřebné knihovny doinstaluje skript automaticky. Testováno na verzi 9.0.1 nástroje `pip`.

Skript umožňuje kompilaci a vytvoření spouštěcího souboru pro daný typ operačního systému. Je zde možnost pro Linux, Windows a Mac. Kompilace je provedena za pomoci knihovny `cx_Freeze`. [\[28\]](#)

Zkompilovaná aplikace je kompatibilní v rámci stejného typu operačního systému na kterém byla kompilace provedena. Aplikace má předpoklady být multiplatformní na základě použití Pythonu 3.6.2 a jeho standartních, volně dostupných knihoven.

Na Windows doporučuji přidat aplikaci do výjimek Windows Defender.

## Repozitář s aplikací

Repozitář obsahuje i subrepozitář se zdrojovými soubory dokumentace. Pro získání veškerého obsahu repozitáře je tedy nutné do terminálu napsat příkaz:

```
git clone -recurse-submodules https://gitlab.com/Sklenar/dupFot.git
```